DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Reference herein to any social initiative (including but not limited to Diversity, Equity, and Inclusion (DEI); Community Benefits Plans (CBP); Justice 40; etc.) is made by the Author independent of any current requirement by the United States Government and does not constitute or imply endorsement, recommendation, or support by the United States Government or any agency thereof.

U. S. Department of Energy
National Energy Technology Laboratory
Federal Grant #: DE-FE0031768

FINAL REPORT

Boiler Health Monitoring Using a Hybrid First Principles-Artificial Intelligence Model

Report No: DOE-WVURC-FE0031768

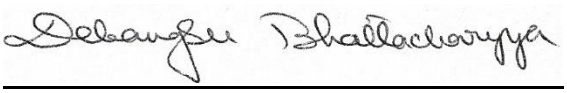
Principle Investigator (PI) and Submitting Official:
Dr. Debangsu Bhattacharyya
Chemical and Biomedical Engineering
West Virginia University
Phone: (304) 293-9335
Fax: (304) 293-4139
Debangsu.Bhattacharyya@mail.wvu.edu

Submission Date: November 30, 2024

UEI: M7PNRH24BBM8

Recipient Organization:
West Virginia University Research Corporation
86 Chestnut Ridge Road
PO Box 6845
Morgantown, WV 26506-6845
Telephone: 304-293-3998
Fax: 304-293-7435

Project/Grant Period: September 1, 2019 – August 31, 2024

Signature of Submitting Official: 

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Table of Contents

TABLE OF CONTENTS **3**

LIST OF FIGURES **7**

LIST OF TABLES **17**

1 INTRODUCTION **21**

1.1 Purpose 21

1.2 Tasks /Approach 22

1.2.1 Task 1.0 Project Management and Planning 22

1.2.2 Task 2.0 Hybrid Model Development, Validation, and Implementation at Plant A

1.2.3 Task 3.0 Validation and Integration of Hybrid Model at Plant A 24

2 DEVELOPMENT AND VALIDATION OF FIRST-PRINCIPLES MODEL AND NONLINEAR STATE ESTIMATOR USING EXTENDED KALMAN FILTER FOR DIFFERENTIAL ALGEBRAIC EQUATION (DAE) SYSTEMS **26**

2.1 Introduction 26

2.2 Dynamic Distributed DAE Model of the Superheater/Reheater 29

2.3 Estimation algorithm: Modified differential algebraic equation-extended Kalman filtering (DAE-EKF) 33

2.4 Case Study: Final Superheater System (Heat Recovery Steam Generator-NGCC Plant B) 36

2.5 Results and Discussion 38

2.5.1 Model and Estimator Validation 38

2.5.2 Sensitivity Studies 42

2.6 Conclusions 49

3 DEVELOPMENT AND VALIDATION OF THE BAYESIAN ML MODEL **50**

3.1 Introduction 50

3.1.1 Development of Sparse Data-Driven Models Amidst Uncertainties 50

3.1.2 Mass Conservation by Sparse Data-Driven Models 51

3.2 Identification of Sparse Static and Dynamic Models from Noisy Data 51

3.2.1 Model Approximation 54

3.2.2 Model Estimation 54
3.2.3 Dynamic Model Selection 58
3.3 Applications of the BIDSAM algorithm 63
3.3.1 Steady state modeling 63
3.3.2 Dynamic Model Building 68
3.3.3 Hybridization with First-principles Model 73
3.4 Mass Conservation by Sparse Data-Driven Models 77
3.4.1 Algorithm I 78
3.4.2 Algorithm II 80
3.5 Case Studies 85
3.5.1 Absorber unit for CO_2 capture plant 85
3.5.2 Results for steady state model 86
3.5.3 Results for dynamic model 89
3.6 Conclusions 92

4 DEVELOPMENT AND VALIDATION OF THE NEURAL NETWORK AND GAUSSIAN RBF MODEL 93
4.1 Introduction 93
4.2 Hybrid All-Nonlinear Series / Parallel Static-Dynamic Neural Networks 94
4.2.1 Literature Review 94
4.2.2 Neural Network Structures 97
4.2.3 Training Algorithms for Hybrid All-Nonlinear Networks 99
4.2.4 Case Study: Dynamic Modeling 104
4.2.5 Comparing All-Nonlinear NN and Bayesian Machine Learning 109
4.2.6 Conclusions 112
4.2.7 Code Availability Statement 112
4.3 Mass-Energy-Thermodynamics Constrained Neural Networks (METCNN) 113
4.3.1 Literature Review 113
4.3.2 Training and Forward Problem for Steady-State / Dynamic METCNN 116
4.3.3 Case Studies and Corresponding Mass Energy Thermodynamics Constraints 123
4.3.4 Results and Discussions: Steady-State Modeling 130
4.3.5 Results and Discussions: Dynamic Modeling 137
4.3.6 Conclusions 140
4.3.7 Code Availability Statement 141
4.4 Gaussian Radial Basis Neural Network (GRAB-NN) Models 141

4.4.1 Literature Review....141
4.4.2 Candidate Network Architectures....145
4.4.3 Training Algorithms....147
4.4.4 Results and Discussions....157
4.4.5 Conclusions....165
4.4.6 Code Availability Statement....165

5 DEVELOPMENT AND VALIDATION OF HYBRID FIRST PRINCIPLES-ARTIFICIAL INTELLIGENCE MODELS FOR BOILER HEALTH MONITORING....166
5.1 Introduction....166
5.2 Components of the Hybrid FP+AI Model....171
5.2.1 Dynamic Distributed DAE Model of the Superheater / Reheater (FP Model) 171
5.2.2 All-Nonlinear Static-Dynamic Neural Networks (NN Model)....173
5.2.3 Bayesian Identification of Dynamic Sparse Algebraic Models (BML Model)....173
5.3 Configurations of Hybrid FP+AI Models....173
5.3.1 Series Hybrid FP+AI Model....174
5.3.2 Integrated Hybrid FP+AI Model....174
5.3.3 Parallel Hybrid FP+AI Model....176
5.4 Description of Industrial Steam Superheater System (Plant A)....178
5.5 Oxide scale growth study for Plant A....180
5.6 Results and Discussion....181
5.6.1 Results from Integrated Hybrid FP+AI Model....182
5.6.2 Results from Integrated-Parallel Hybrid FP+AI Model....194
5.6.3 Results from Oxide Scale Growth Modeling....200
5.7 Conclusions....203

6 VALIDATION AND INTEGRATION OF HYBRID MODEL AT PLANT A....204
6.1 Overview on Boiler Health Management....204
6.2 Demonstration at Plant A....206
6.2.1 Value of a live demonstration....206
6.2.2 Reason for selecting this plant/unit....206
6.2.3 Location in the Superheater....208
6.2.4 Dutchmen Design and Fabrication....208
6.2.5 Installation....209
6.2.6 Analysis of Operational Data....209

6.2.7 Flue Gas Survey at Plant A 214
6.3 Metallography work 219
6.3.1 AUSC Steam Loop from 2011 219
6.3.2 Analysis of Ex-Service 347H Tubes from Plant A 225
6.3.3 Analysis of T22 Dutchmen 228
6.3.4 Discussion on Temperature Exposure and Correlation of Various Indicators, etc. 243
6.4 Gas-Touched Length Analysis (GTLA) 245
6.4.1 Outline of GTLA Model 246
6.4.2 Development of GTLA Model 246
6.5 Thermal Conductivity Tests 251
6.5.1 Heat Transfer Sensitivity to Oxide Conditions 251
6.5.2 Existing Oxide Thermal Conductivity Values in Literature 251
6.5.3 Laser Flash Diffusivity Testing of Ex-Service Tubes 252
6.5.4 Water Bath Heat Flux Experiment 257
6.5.5 Recommended values for WVU model and beyond 261
6.6 Enhanced Life Assessment Tools 262
6.6.1 Existing Tools 262
6.6.2 Survey of EPRI Members and their Future Needs 263
6.6.3 Proposal Framework for Superheater Damage Evaluation Software 267

7 RECOMMENDATIONS AND RESEARCH OUTPUT 277
7.1 Research Output 277
7.1.1 Journal Publications 277
7.1.2 Conference Presentations 279
7.1.3 Public GitHub Repositories 281

APPENDICES 282
Appendix A: Additional validation results for the first principles model and nonlinear estimator for Plant B superheater system 282
Appendix B : Additional results for the development and validation of the Bayesian ML Model 290
Appendix C: Additional results for the development and validation of the Neural Network and Gaussian RBF Models 307
Appendix D : Superheater Damage Evaluation Software Specification Document 317

REFERENCES 365

List of Figures

Fig. 2.1. Module representation of the superheater/reheater distributed model 29

Fig. 2.2. Schematic representation of the final superheater system used in this study – locations identified on the left are at the elevation (in ft) indicated. Steam exiting the outlet header is transmitted through a main steam pipe to a steam turbine.....36

Fig. 2.3. Representation of the six-pass final superheater system using control volumes (the second index number in each control volume refers to the tube pass number for the superheater system).....37

Fig. 2.4. Input variables considered for the industrial data. (a) Flue gas inlet temperature ($T_{fg,in}$), (b) Steam inlet temperature ($T_{st,in}$), (c) Flue gas mass flowrate (mfg), (d) Steam mass flowrate (mst), (as shown in Fig. 2.3).....38

Fig. 2.5. Comparison of the measured data of the main steam outlet temperature with the uncorrected-model-prediction and estimator-corrected results.....39

Fig. 2.6. Comparison of the model and estimator results with the measured data for the flue gas temperature at an elevation of 47 feet (G 47').....40

Fig. 2.7. Comparison of the model and estimator results with the measured data for the tube temperature at an elevation of 75 feet (E 75').....41

Fig. 2.8. Estimation of the unmeasured flue gas outlet temperature ($T_{fg,out}$, as shown in Fig. 2.3).....42

Fig. 2.9. Natural gas feed profile for the sensitivity study.....43

Fig. 2.10. Profiles of boundary variables for the sensitivity study. (a) Flue gas inlet temperature ($T_{fg,in}$), (b) Steam inlet temperature ($T_{st,in}$), (c) Flue gas mass flowrate (mfg),(d) Steam mass flowrate (mst), (as shown in Fig. 2.3).....44

Fig. 2.11. Comparison of the true values of the main steam outlet temperature ($T_{st,out}$, as shown in Fig. 2.3) with the noisy data, model results, and estimator results. 45

Fig. 2.12. Comparison of the true values of the flue gas outlet temperature ($T_{fg,out}$, as shown in Fig. 2.3) with the noisy data, model results, and estimator results. 46

Fig. 2.13. Estimated average steam temperature profile in the horizontal direction (x-direction, as shown in Fig. 2.3) along the superheater.....47

Fig. 2.14. Estimated average flue gas temperature profile in horizontal direction (x-direction, as shown in Fig. 2.3) along the superheater.....47

Fig. 2.15. Estimated Pass 1 (near steam outlet) tube temperature profiles in vertical direction (y-direction, as shown in Fig. 2.3).....48

Fig. 2.16. Estimated Pass 6 (near steam inlet) tube temperature profiles in vertical direction (y-direction, as shown in Fig. 2.3).....49

Fig. 3.1.Decision tree for model selection.....59

Fig. 3.2.Monotonic decrease in cost function down a branch 60

Fig. 3.3. Detailed flowchart for the Bayesian Identification of Dynamic Sparse Algebraic Model (BIDSAM) algorithm.....62

Fig. 3.4. Schematic diagram of non-isothermal van de Vusse reactor 63

Fig. 3.5. Validation plots for C_B (a) BIDSAM model (b) ALAMO A model (c) ALAMO B model. 67

Fig. 3.6.Comparison of models identified from true and noisy data (a) BIDSAM (b) ALAMO A (c) ALAMO B. 67

Fig. 3.7. Plot of magnitudes of model coefficients (a) BIDSAM (b) ALAMO. 68

Fig. 3.8. Absorber unit for CO_2 capture 69

Fig. 3.9. Predictions of (a) flue gas outlet temperature (b) flue gas CO_2 concentration without re-initialization. 70

Fig. 3.10. Predictions of (a) flue gas outlet temperature (b) flue gas CO_2 concentration with re-initialization required. 71

Fig. 3.11. Predictions of (a) flue gas outlet temperature (b) flue gas CO_2 concentration for series of step changes in flue gas flow rate. 73

Fig. 3.12. Validation of first-principles model (a) prediction of $T_{st,\,main}$; (b) prediction error.....74

Fig. 3.13. Structure of hybrid model for heat transfer coefficient corrective term. 75

Fig. 3.14. Validation of hybrid model (a) prediction of $T_{st,\,main}$; (b) prediction error.....75

Fig. 3.15. Structure of hybrid model for predicting individual tube temperatures.....76

Fig. 3.16. Validation of hybrid model predicting steam temperatures at specific tube locations.....76

Fig. 3.17. (a) Block diagram for the MCBIDSAM model development using Algorithm I; (b) Tearing Approach for Algorithm I.....78

Fig. 3.18. Algorithm I for solving inverse problem for MCBIDSAM model. 80

Fig. 3.19.Algorithm II for solving inverse problem for MCBIDSAM model. 81

Fig. 3.20. Block diagram of a system with 2 inputs and 4 outputs. 81

Fig. 3.21.Schematic of CO_2 absorber unit 86

Fig. 3.22.Validation plots for model developed from noisy data using Algorithm II: (a)-(b) BIDSAM, (c)-(d) MCBIDSAM, (e) absolute percentage error in atom balance. 88

Fig. 3.23. Validation plots for the prediction of $L_{CO\ out}$ with models developed from noisy data with bias (a) Algorithm I (b) Algorithm II. 90

Fig. 3.24.Absolute percentage atom imbalance for validation of models for CO_2 absorber unit developed from noisy data with bias (a) Algorithm I (b) Algorithm II. 91

Fig. 4.1. Architectures of fully nonlinear networks: (a) NLS-NLD, (b) NLD-NLS, and (c) NLS||NLD 98

Fig. 4.2. Key to proving equivalence of optimization problems representing (a) simultaneous and (b) sequential approaches 99

Fig. 4.3. Algorithm for Training Hybrid Series NLD – NLS Network Model....100

Fig. 4.4. Algorithm for Training the Hybrid Parallel NLS||NLD Network Model 102

Fig. 4.5. Schematic of the absorber decoupled from the CO_2 capture pilot plant flow diagram 105

Fig. 4.6. Comparison between model prediction for validation data for the tower outlet clean flue gas flowrate for (a) NLS-NLD, (b) DABNet, (c) LSTM2 and (d) GRU2 networks....108

Fig. 4.7. Comparison between predictions obtained from the optimal NN and BML models for flue gas outlet temperature 110

Fig. 4.8. Generic block-oriented configuration for steady-state METCNN models 117

Fig. 4.9. Generic architectures of dynamic (hybrid parallel) METCNN models in (a) absence and (b) presence of system holdup measurements....121

Fig. 4.10. Schematic of the (a) lumped and (b) distributed superheater / reheater model for ECNN implementation....124

Fig. 4.11: Process flowsheet of acetic anhydride production for METCNN implementation 126

Fig. 4.12: Comparison of results between ECNN and NN w/o energy constraints for the simulation data of TSt, out 131

Fig. 4.13. Comparison of minimum % RMSE for simulation data using ECNN from sensitivity analyses based on (a) sampling and (b) availability of local measurements....133

Fig. 4.14. Comparison of results between METCNN and NN w/o mass-energy-thermodynamics constraints (METC) for simulation data of reactor outlet temperature $Tout$ 135

Fig. 4.15 . Comparison between METCNN, MECNN and NN w/o METC in terms of violating (a) carbon (C), (b) hydrogen (H), (c) oxygen (O) and (d) energy (Q) balance constraints during simulation....136

Fig. 4.16. Results from dynamic (hybrid parallel) ECNN for TSt, out distribution in spatial coordinates during (a) training and (b) simulation....138

Fig. 4.17. Comparison of results between dynamic METCNN, dynamic MECNN, and hybrid series NN w/o METC (a) for simulation data of $Tout$ and (b) in terms of violating energy (Q) balance constraints during simulation 139

Fig. 4.18. General architecture of the hybrid GRAB-NN model.....146

Fig. 4.19. Flowchart describing the rigorous approach for parameter estimation of RBF network 150

Fig. 4.20. Network configuration for dynamic McRBF / McGRAB-NN models 156

Fig. 4.21.: Results from the optimal unconstrained RBF model with probabilistic centers obtained through the rigorous training algorithm for validation data of (a) CA and (b) CB 158

Fig. 4.22. Variation in average % MSE for all possible combinations of ANN and RBF hidden nodes in the hybrid GRAB-NN model for validation data $(nh = 15)$ 161

Fig. 4.23.: Comparison of results between McRBF and unconstrained RBF models with respect to true and measured validation data of CB.....163

Fig. 4.24.: Comparison between McRBF and unconstrained RBF models in terms of violating (a) carbon (C) and (b) hydrogen (H) atom balance constraints during validation.....163

Fig. 4.25.: Comparison of results between dynamic McRBF and unconstrained RBF models with respect to true and measured validation data of CB 164

Fig. 5.1. Control volume block of the dynamic first principles superheater model. 171

Fig. 5.2. Block-oriented configuration of series hybrid FP+AI model during training and simulation. 174

Fig. 5.3. Block-oriented configurations of integrated hybrid FP+AI model during (a) training and (b) simulation. 175

Fig. 5.4. Block-oriented configurations of parallel hybrid FP+AI model during (a) training and (b) simulation. 176

Fig. 5.5. Block-oriented configurations of (a) series-parallel and (b) integrated-parallel hybrid FP+AI models during simulation. 177

Fig. 5.6. Final superheater system (Plant A) layout. Note: the actual tube arrangement involves smooth transitions rather than the angular transitions shown in the schematic diagram. 179

Fig. 5.7. Final superheater system layout using control volumes (second index of individual control volumes represents actual pass number of the superheater system) for Plant A. 179

Fig. 5.8. Dynamic profiles of input variables for Plant A final superheater system considered for standalone FP model, i.e., (a) steam inlet temperature Tst, in, (b)

steam mass flowrate (mst), (c) flue gas inlet temperature Tfg,in, and (d) flue gas mass flowrate (mfg) (see Fig. 5.7) 183

Fig. 5.9..Temporal profiles of the plant load among low, medium, and high values for the superheater system at Plant A. 183

Fig. 5.10. Comparison of results from the standalone FP model with the measured data for main steam outlet temperature with respect to Plant A. 184

Fig. 5.11. Comparison of results between standalone FP vs hybrid (a) FP+NN and (b) FP+BML models for main steam outlet temperature with respect to the superheater system at Plant A. 186

Fig. 5.12. Main steam outlet temperature validation using hybrid grey-box modeling (FP +NN) for 22 months using training data for 10 days 188

Fig. 5.13. Main steam outlet temperature validation using hybrid grey-box modeling (FP +BML) for 22 months using training data for 10 days 189

Fig. 5.14. Adaptive Hybrid Modeling algorithm 190

Fig. 5.15. Main steam outlet temperature validation using hybrid grey-box modeling (FP +NN) for 22 months using adaptive training 192

Fig. 5.16. Main steam outlet temperature validation using hybrid grey-box modeling (FP +BML) for 22 months using adaptive training 193

Fig. 5.17. Distribution of transient steam outlet temperature profiles for Plant A, across the flow circuits/locations along the width direction 194

Fig. 5.18. Comparison of results between hybrid integrated-parallel (a) FP+NN and (b) FP+BML models for predicting steam outlet temperatures along width direction with respect to Plant A. 197

Fig. 5.19. Comparison of results between hybrid integrated-parallel FP+NN models for predicting steam outlet temperatures along width direction with respect to Plant A for 22 months using adaptive training 199

Fig. 5.20. Adaptive Hybrid Integrated Parallel Dutchmen Steam temperatures result for 22 months of data. 201

Fig. 5.21. Comparison of Oxide scale growth for the 5 dutchmen locations for the 22 months period 202

Fig. 6.1. Schematic diagram of the configuration of tube banks in the Plant A boiler including the location of the added dutchmen as a hatched box in the final superheater (7). White arrows show the direction of steam flow. Red and yellow arrows show the flow of flue gas. 207

Fig. 6.2. Excerpt of the BTA drawing for T22 dutchmen to be installed in Plant A. Dimensions in inches 208

Fig. 6.3. Boxplots of temperature distribution across the width of the final superheater over a 1-month timespan. Green is inlet temperatures, blue is the mean temperatures, and yellow is the outlet temperatures. 209

Fig. 6.4. Peak temperatures reported for TC locations in Plant A superheater for each month March 2022 through January 2023. The colors range from dark grey (525°C) to dark yellow (615°C) and white is no data or below 525°C. 212

Fig. 6.5. An approximation of aggregated damage for TC locations in Plant A superheater for each month March 2022 through January 2023. The color scheme is a relative index of damage, not quantitative. Dark blue indicates the least damage, dark yellow the most. White indicates no data. 212

Fig. 6.6. Collection of normalized histogram plots showing the thermocouple data at the locations of interest. 213

Fig. 6.7. A schematic diagram illustrating the HVT lance system used to measure flue gas temperatures and velocities in the superheater region of Plant A (reproduced from https://www.stormeng.com/ high-velocity-thermocouple-kit/). 214

Fig. 6.8. Schematic of Plant A and the access port just upstream of the final superheater. 216

Fig. 6.9. Boxplot of the temperature data recorded by HVT at Plant A during December 2021. 216

Fig. 6.10. Steam temperature measurements from the inlet header (grey) and outlet headers (outer row in blue and inner row in orange) after the 2021 outage and at low unit load. 217

Fig. 6.11. Steam temperature measurements from the inlet header (grey) and outlet headers (outer row in blue and inner row in orange) before the 2021 outage and at low unit load. 218

Fig. 6.12. Life consumption based on 10,000 hours at the temperatures described in Fig. 6.10 219

Fig. 6.13. Schematic showing the locations of Super304H segments in Plant A steam loop. 221

Fig. 6.14. Image showing the locations where Super304H segments from the AUSC steam loop were sectioned with respect to the flue gas flow and their appearance of the section after polishing. 221

Fig. 6.15. Microstructure of Super304H after electrolytic etching with 20% KOH solution at 3V showing the presence of sigma phase. 222

Fig. 6.16. Image of Super304H microstructure (left) and the result obtained after segmentation using image analysis sub-routine which highlights sigma phase showing about 2% sigma. 222

Fig. 6.17. Graphs showing the distribution of sigma phase from OD to ID regions of the six Super304H sections. 223

Fig. 6.18. Thermodynamic predictions from ThermoCalc of Super 304H phase stability with temperature. Note the predicted volume% of sigma with temperature. 224

Fig. 6.19. Summary of ID oxide scale measurements taken from Tube 51. Inner refers to spinel layer and outer refers to magnetite or magnetite plus hematite. 226

Fig. 6.20. Summary of ID oxide scale measurements taken from Tube 99. Inner refers to spinel layer and outer refers to magnetite or magnetite plus hematite. 226

Fig. 6.21. Summary of ID oxide scale measurements taken from Tube 147. Inner refers to spinel layer and outer refers to magnetite or magnetite plus hematite. 226

Fig. 6.22. Representative optical micrograph of ID scale on Tube 51 227

Fig. 6.23. Representative optical micrograph of ID scale on Tube 99 227

Fig. 6.24. Representative optical micrograph of ID scale on Tube 147 227

Fig. 6.25. As-received image of one of the six fabricated dutchmen tubes showing the central T22 portion and the two 347H safe-ends. The safe ends were welded to the T22 tubing to facilitate installation in Plant A. 228

Fig. 6.26. a and b: Images of the two sections removed from the provided dutchmen showing the length of T22 tube, the length of 347H safe-end and the weld fabricated between these tubes. Cross-sections of the T22 tubes were taken from these welded sections as marked by the dashed lines. 229

Fig. 6.27. A scanning electron microscope image of the T22 substrate near the tube ID regions showing the ID surface which had not clear evidence of any scale present in the as-fabricated state. 229

Fig. 6.28. Image of the as-received ex-service dutchmen tubes from Plant B. 230

Fig. 6.29. a) Schematic of the T22 dutchmen tubes received showing 347H safe ends, and b) image showing ~24 inches long sections from the T22 dutchmen removed from one end of the as-received tubes. 231

Fig. 6.30. Image of the T22 cross sections taken from the five tubes for microhardness testing (full ring sections) and ID oxide scale analysis (cut into quadrants). 231

Fig. 6.31. a) Microhardness distribution map of T22 tube in as-manufactured condition and, b) image showing ~24 inches long sections from the T22 dutchmen removed from one end of the as-received tubes. 0° direction is up. 232

Fig. 6.32. A box plot showing the distribution of microhardness in the five ex-service T22 tubes in comparison with the hardness of T22 steel in the as-manufactured condition. 233

Fig. 6.33. Representative optical images of steam-side oxide scales captured at 500X magnification from a) Tube 11 and b) Tube 147. 234

Fig. 6.34. Images showing a) an as-captured image of the steam-side oxide scale from Tube 189 and, b) after image post-processing operation where scale thickness were measured at multiple locations along the width of the image 234

Fig. 6.35. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 11 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 11. 235

Fig. 6.36. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 51 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 51. 236

Fig. 6.37. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 99 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 99. 237

Fig. 6.38. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 147 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 147. 238

Fig. 6.39. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 189 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 189. 239

Fig. 6.40. Box plot showing the distribution of oxide scale thickness across the circumference of the five tubes studied in this work. 240

Fig. 6.41. Box plot with the distribution of steam-side oxide scale thickness measurements from the five T22 tubes. 240

Fig. 6.42. A plot of tube test results showing average hardness and average rupture time. 242

Fig. 6.43. Schematic diagram illustrating the distribution of temperature across the tube wall from the flue gas region (left) to the center of the tube (right). 246

Fig. 6.44. Schematic showing a discretized tube element of length ΔL, where ‘A’ is the steam inlet plane and plane ‘B’ is the steam outlet plane. Radial distribution of temperature along plane ‘A’ of such a discretized element is calculated in the GTLA model. 247

Fig. 6.45. Schematic of a transverse section of a tube wall showing various interfaces present in the radial direction (A, B, C, D and E) where the steady-state temperature can be calculated. 249

Fig. 6.46. Predicted temperature profile along the AUSC steam loop under conditions described in Table 6-7. 250

Fig. 6.47. Survey of thermal conductivity of metal oxides reported in the literature [400] 252

Fig. 6.48. *Representative optical metallography of ex-service tube samples A (Stage 3), B (Stage 3), C (Stage 3), D (Stage 3), E (Stage 2), and G (Stage 3). (These images are not all to the same scale.)* 256

Fig. 6.49. Simplified Historical metal oxide thermal conductivity data with new test data (CMC) overlayed. T91 samples from “C”, with a thinner lower porosity oxide are

shown as open symbols and show a more consistent conductivity than thicker, more porous data. 257

Fig. 6.50. Photograph (left) showing the experimental setup for the water bath heat flux testing at the EPRI laboratories. The associated schematic diagram (right) identifies key parts of the equipment. 258

Fig. 6.51. Schematic representation of typical temperature profile illustrated across one half the apparatus which develops during the heat flux experiment. This figure also highlights the approximate location of attached thermocouples in EPRI's implementation. 258

Fig. 6.52. Example plot of heat flux experiment data, this data from Tube B (with the bubbler on) 260

Fig. 6.53. Cumulative probability of failure versus time from a Monte Carlo simulation of T91 tubing under coal boiler superheater conditions after 200,000 hours of service, run by TULIP 2.0 [405] 263

Fig. 6.54. Schematic of major building blocks associated with the development of software like the tool described in Appendix D 271

Fig. 6.55. An early mock-up of the graphical user interface which the user could use to select the desired analysis. 274

Fig. 6.56. A proposed timeline of software development activities and project duration for the tool and analyses described in Appendix D. 275

Fig. A. 1.Comparison of the model and estimator results with the measured data for the flue gas temperature at an elevation of 71 feet (G 71'). 282

Fig.A. 2.Comparison of the model and estimator results with the measured data for the tube temperature at an elevation of 49 feet (E 49'). 283

Fig.A. 3.Comparison of the model and estimator results with the measured data for the tube temperature at return bend 2 (RB 2). 283

Fig.A. 4.Comparison of the model and estimator results with the measured data for the tube temperature at return bend 1 (RB 1). 284

Fig.A. 5. Comparison of the model and estimator results with the measured data for the tube temperature an elevation of 6 feet (E 6'). 285

Fig.A. 6. Estimation of unmeasured intermediate headers outer tube wall temperatures (a) Intermediate Header 2 outer tube wall temperature (b) Intermediate Header 3 outer tube wall temperature (see Fig. 2.2). 287

Fig.A. 7. Estimation of unmeasured intermediate headers flue gas temperatures. (a) Intermediate Header 2 flue gas temperature (b) Intermediate Header 3 flue gas temperature (see Fig. 2.2). 288

Fig.A. 8.Estimated outer tube wall temperature profiles along the intermediate headers and return bends (see Fig. 2.2). 289

Fig.A. 9 .Estimated flue gas temperature profile along the intermediate headers and return bends (see Fig. 2.2). 289

Fig. B. 1. PRBS signal for generating dynamic data for Van de Vusse reactor. 292

Fig. B. 2. Transients in all key input variables during step tests in the flue gas flowrate for CO2 absorber. 293

Fig. B. 3. Lean solvent inlet flow rate as input variable (a) raw plant measurement (b) denoised data using k-nearest neighbor algorithm. 293

Fig. B. 4. Simulation plots for models trained from single step change in flue gas flow rate (a) noise removed from data (b) noise not removed from data. 294

Fig. B. 5. Simulation plots for models trained from data for multiple step changes in flue gas flow (a) SINDy A1 model (b) zoomed portion of SINDy A1 model validation. 295

Fig. B. 6. Simulation plot for SINDy A1 model with all successive predictions based on measurements (b) zoomed portion of the validation plot. 295

Fig. B. 7. Schematic diagram of isothermal CSTR. 298

Fig. B. 8. Validation plots for model developed from noisy data using Algorithm II: (a)-(b) without mass constraints, (c)-(d) with mass constraint. 302

Fig. B. 9. Validation plots for model developed from noisy data using Algorithm I: (a)-(b) without mass constraints, (c)-(d) with mass constraint. 303

Fig. B. 10. Input variables for validation data. 304

Fig. B. 11. Validation plots for model developed from noisy data using Algorithm I: (a)-(b) without mass constraints, (c)-(d) with mass constraint. 305

Fig. C. 1. Comparison between the (a) NLS-NLD, (b) DABNet, (c) LSTM2, and (d) GRU2 for the validation data for the flue gas outlet temperature 308

Fig. C. 2. Comparison between the (a) NLS-NLD, (b) DABNet, (c) LSTM2, and (d) GRU2 for the validation data for CO_2 concentration in the outlet flue gas stream 309

Fig. C. 3. Comparison of prediction results obtained from NN and BML models for (A) flue gas outlet flow rate and (B) flue gas outlet CO_2 concentration 309

Fig. C. 4. Training algorithm for dynamic (hybrid parallel) METCNN model in absence of system holdup measurements 311

Fig. C. 5. Comparison of error between ECNN and NN w/o energy constraints for energy balance at intermediate grids during simulation 312

Fig. C. 6. Comparison of results between METCNN and NN w/o METC for simulation data of outlet concentration of A (CA) 313

Fig. C. 7. Comparison between METCNN, MECNN and NN w/o METC in terms of violating (a) carbon (C), (b) hydrogen (H), (c) oxygen (O) and (d) energy (Q) balance constraints during training 314

Fig. C. 8. Results from the optimal unconstrained RBF model with deterministic centers obtained through the approximate training algorithm for validation data of (a) CA and (b) CB 314

Fig. C. 9. Results from the optimal unconstrained RBF model with probabilistic centers obtained through the approximate training algorithm for validation data of (a) CA and (b) CB 315

Fig. C. 10. Results from optimal hybrid GRAB-NN model for validation data of (a) CA and (b) CB 315

Fig. C. 11. Comparison between dynamic McRBF and unconstrained RBF models in terms of violating (a) carbon (C) and (b) hydrogen (H) atom balance constraints during validation 316

List of Tables

Table 2.1: RMSE for three measured variables for the final superheater system (NGCC plant) 40

Table 3.1.Van de Vusse reactor parameters 65

Table 3.2. Performance measures of BIDSAM and ALAMO models (trained using true data and noisy data) with respect to true data 66

Table 3.3. Performance indices of identified models for the single step change in the flue gas flow rate 71

Table 3.4. Performance indices of identified models for series of step changes in flue gas flow rate. 73

Table 3.5. Performance measures of the steady state models of the absorber unit of CO_2 capture plant. 87

Table 3.6. Performance measures of the dynamic models of the absorber unit of CO_2 capture plant. 91

Table 4.1: Comparison among the training algorithms, computational time and percentage MSE for both training and validation data of various types of models for the dynamic CO_2 capture pilot plant system w.r.t Objective 1 and 2 106

Table 4.2: Comparison in MAIE between NLS-NLD (optimal for this case study) and other state-of-the-art models for all outputs variables w.r.t. the dynamic CO_2 capture pilot plant system.....107

Table 4.3: Comparison among the optimal hybrid model and existing state-of-the-art models in terms of MSE during training and corresponding AICc values for CO_2 capture pilot plant.....108

Table 4.4: Comparison between NN and BML models in terms of overall RMSE / MAIE for the validation data, training computational time and number of parameters 110

Table 4.5. List of candidate thermodynamics models considered for the PFR system to explicitly express $\mathrm{hex(t)}$ as a function of Tin and Tout.....128

Table 4.6. Comparison among different candidate thermodynamics models imposed as constraints in METCNN in terms of percentage RMSE for training and simulation data of $Tout$.....134

Table 4.7. Performance comparison between optimal unconstrained RBF models with deterministic / probabilistic centers using rigorous and approximate approaches for validation data of CA and CB 159

Table 5.1. List of model input and output variables in the integrated hybrid FP+AI models for the superheater system of Plant A.....185

Table 5.2. Comparison between hybrid integrated FP+NN and FP+BML models in terms of overall prediction RMSE, MAIE, number of parameters and training computational time for the main steam outlet temperature of Plant A 186

Table 5.3 .List of model input and output variables in the integrated-parallel hybrid FP+AI models for the superheater system of Plant B 196

Table 5.4. Comparison between FP+NN and FP+BML models in terms of RMSE, MAIE, and computational expense for predicting steam outlet temperatures in Plant A at individual flow circuits along the width direction.....197

Table 5.5. Estimates of oxide scale thickness from the adaptive hybrid model and the maximum and minimum values of measured steam-side oxide scale thicknesses of the four tubes. 203

Table 6-1: Summary of hardness data from the microhardness testing of the six tested T22 steels.....233

Table 6-2: Summary of steam-side oxide scale thickness measurements from the five T22 tubes. 240

Table 6-3: Estimates of tube metal temperature calculated from the mean and the maximum values of measured steam-side oxide scale thicknesses of the five tubes.....241

Table 6-4: Duration of creep testing performed on ex-service T22. All tests performed at 600°C and 90 MPa. 242

Table 6-5: Summary of data and sample analysis for each of the five locations.....244

Table 6-6: Ranking of select data from Table 6-5 for each of the five dutchmen locations 245

Table 6-7: Example operation parameters evaluated for GTLA of the AUSC Steam Loop representing the range of observed conditions. 250

Table 6-8: Thermophysical properties of ex-service tube materials as determined by laser flash diffusivity testing by the external contractor. These thermophysical property values were used in the calculation of oxide thermal conductivity as determined by laser flash diffusivity and presented in Table 6-9. 253

Table 6-9*: A summary of laboratory laser flash diffusivity testing results on the ex-service tube specimens.* 255

Table 6-10: Geometric characteristics of ex-service tubes tested in thermal flux experiments.....259

Table 6-11: Geometric characteristics of ex-service tubes tested in heat flux “reverse” water bath experiments 261

Table A.1: RMSE for five measured variables for the final superheater system (NGCC plant) 286

Table B. 1. Selected basis functions for BIDSAM and ALAMO steady state models for non-isothermal Van de Vusse reactor.....290

Table B. 2. Training time analyses for developing steady state model for van de Vusse reactor. 291

Table B. 3. Selected basis functions for some of the estimated models for non-isothermal Van de Vusse reactor. 292

Table B. 4. Performance indices of identified models for the step changes in the flue gas flow rate for CO2 absorber. 294

Table B. 5. Selected basis functions for BIDSAM models for multiple step changes in flue gas flow rate. 296

Table C. 1. Training time for different solution combinations for NLS-NLD, NLD-NLS and NLS||NLD for the NCCC pilot plant for CO_2 capture.....307

Table C. 2. Optimal parameter values of NN and BML models for the CO_2 Capture Pilot Plant.....310

Table D. 1. Data Required for Coal Boiler Analyses.....347

Table D. 2. Geometry Required for Coal Boiler Analyses.....348

Table D. 3. Material Properties Required for Coal Boiler Analyses.....349

Table D. 4. Data required for HRSG Analyses 350
Table D. 5. Geometry required for HRSG Analyses 351
Table D. 6. Material Properties required for HRSG Analyses 352

1 Introduction

1.1 Purpose

Due to increased penetration of the intermittent renewables to the grid, most PC plants are being forced to cycle their loads frequently and rapidly, operate at low load condition for sustained period, and start up and shut down several hundred times in a year. These severe operations are causing substantial damage to the boiler components compromising the reliability of these PC plants. An online health monitoring tool can be instrumental in understanding the impacts of load-following and can eventually help the PC plants to develop advanced process control strategies for improved flexibility without compromising safety and reliability.

While first-principles physics-based models can be predictive, it is still difficult to capture the complex phenomena in the boiler components by these models. Furthermore, these models are computationally expensive and difficult to adapt online. On the other hand, artificial intelligence (AI) models can be simpler to construct and adapt online, but may not be predictive especially when they are extrapolated and/or if the data used for developing the AI models suffer from information gap. The key strengths of the physics-based and AI models can be exploited by synergistically coupling them to develop a hybrid model. Since neither the hybrid model nor the online data are complete and accurate, they can be leveraged using an optimal quantitative estimation algorithm to yield a boiler health-monitoring framework that is adaptive, real-time, and accurate.

Therefore, this project seeks to develop methodologies and algorithms to accomplish: (i) a hybrid first-principles-AI model of the PC boiler, (ii) physics-based approach to material damage informed by ex-service component evaluation, (iii) a transformative, online health-monitoring framework that synergistically leverages the hybrid models and plant measurements to provide the spatial and temporal profile of key transport variables and characteristic measures for plant health, and (iv) a field implementation and demonstration at Southern Company's Plant A .

The methodologies and algorithms developed in this project will be first calibrated and validated using the available data from Southern Company's plants. The framework will also be deployed at Plant A specifically and evaluated for online monitoring of the boiler health.

The Electric Power Research Institute's (EPRI) objective as sub-awardee with WVU is to provide real-world operation and material damage inputs to the hybrid creep and thermo-mechanical fatigue damage models. EPRI's existing relationship with utilities operating coal boilers, specifically Southern Company who has agreed to support this project, will afford the opportunity to collect operational data and ex-service (e.g. damaged) material. EPRI is uniquely positioned in the industry to liaise this type of experiential validation work not only to WVU, but through EPRI's internal metallographic characterization laboratory; there, this scope can be cross referenced with previous work on damaged ex-service boiler components. Further, EPRI has existing physics-based models (e.g., TULIP) that support predictive analysis of damage accumulation within a boiler. Currently these software packages are used by utilities off-line. This project will develop AI models which provide inputs for creep lifing calculations for tube life methods such as EPRI's TULIP and facilitating deployment of these powerful software for online monitoring.

Given the operational data from the test loop and advanced microstructural characterization, the hybrid model can learn from real world conditions. With the assistance of Southern Company, EPRI and WVU will install appropriate sensors to an operating coal boiler, monitor a particular component of interest, and compare its degradation over the duration of this project to the predictions of the model. That opportunity to demonstrate and validate will bring the proposed hybrid damage monitoring framework to TRL5. Bringing this technology to TRL6 would require fully integrating the approach into the plant monitoring and reporting systems in Plant A or another facility which is outside of the scope of this project.

The approach/framework proposed in this project is generic and can be readily applied online for health monitoring of other PC plants and superheater/reheater section of natural gas combined cycle (NGCC) plants. Thus, a key goal of the proposed project is to develop the framework so that it is easy to understand and implement for broader acceptability by and applicability to large number of power plants.

1.2 Tasks /Approach

The section below has been organized by Task (each subsection), and the work completed and summary of the results will be contained within each subsection. Only the Tasks that were planned to be focused upon within the reporting quarter will be presented. The following subsections and work were planned for this quarter:

1.2.1 Task 1.0 Project Management and Planning

This task will include all work elements required to maintain and revise the Project Management Plan, and to manage and report on activities in accordance with the plan. It will also include the necessary activities to ensure coordination and planning of the project with DOE/NETL and other project participants.

1.2.2 Task 2.0 Hybrid Model Development, Validation, and Implementation at Plant A

Key Responsibility: WVU
The WVU team will develop/adapt the first-principles and AI models for the boiler of the coal-fired power plant, validate these models with the data from the test loop and implement these models online at Plant A for evaluation.

Subtask 2.1 Plant Data Evaluation

The data obtained from Plant A will be evaluated for completeness and sorted for model development and validation. This step will also include identification of the representative data from the three-year window to be used for model calibration and validation.

Subtask 2.2 Adapting the First-Principles Model to Plant A and B

The available supercritical pulverized coal (SCPC) power plant model at WVU will be adapted to the plant A and B host site boiler. First, the Plant B system will be modeled. An extensive instrumentation is available in Plant B superheater section that provides the steam flowrates, flue gas and tube temperatures at the front row of the tubes that are generally subjected to the highest thermal stresses. Eventually this model will be updated for the final demonstration site, i.e. superheater section in Plant A.

The existing dynamic model at WVU for the superheater/reheater section of a furnace is a 2-d model for overall crossflow configuration with due consideration of flow passes. In addition, the dynamics of the tube wall temperature are provided by the model. Steam properties are represented by IAPWAS R7-97. The model also includes additional external and internal heat transfer resistances due to deposit/scaling, but they are currently user-provided. These will be estimated by using the Bayesian ML model. The first-principles model will later be modified for the existing superheater/reheater component that will be monitored as part of this project. The exact component and its location that will be monitored online will be identified as part of the project depending on the availability of the data and accessibility of information from the host site.

Subtask 2.3 Development and Validation of the Bayesian ML Model

The existing Bayesian ML technique will be modified with due consideration of plant data and applicability to real life. While computational expense is a critical criterion for real-life application, one important thing that needs to be considered is generalization capability of the developed model, i.e. avoid overfitting. Tradeoffs between sparsity and training error will be considered for better generalizability of the model by maximizing the Kullback-Leibler information (i.e. minimizing the Kullback-Leibler discrepancy between the true probability density function and the likelihood of the model). While the Bayesian ML approach inherently considers noise in the data in the EM algorithm, this aspect will be visited. In presence of high noise, if exists, the cost function in the EM algorithm will be suitably modified by including a loss function such as Huber's function or Hampel's tanh estimator. Since the output of this model, i.e. inner and outer heat transfer resistance profiles, cannot be validated independently, it will be jointly validated with the first-principles model first using the test-loop data and later using the online data.

Subtask 2.4 Development and Validation of the Gaussian RBF Model

Input data for TULIP will be generated from the combined first-principles/Bayesian ML model. The output data from TULIP will then be used to train the Gaussian RBF model that is a probabilistic neural network but has universal approximation and regularization capabilities. The training will be conducted offline, but repeated, when needed, based on the update of TULIP as part of this project. The training will be conducted using the Bayes learning rule where the model size and parameters, noise parameters, and regularization parameters are treated as unknown random variables. Given sufficient training data collected from TULIP, the EM algorithm should asymptotically converge to a lower bound on the generalization error. Similar to Subtask 2.3, tradeoffs between sparsity and training error will be considered for selecting the final model. The model will be validated through *k*-fold cross-validation.

Subtask 2.5 Modification and Implementation of the Optimal DAE Estimator

The optimal DAE estimator that receives the measurement data and uses the first-principles model with the embedded ML model will be developed by suitably modifying the existing algorithm at WVU. For making the estimator suitable for online application, dimensionality of the first-principles model may need to be reduced and approximations of the algebraic equations especially for the properties model may need to be done. Error versus computational expense of linear vs nonlinear estimators will also be evaluated. This task will also evaluate if any data preprocessing algorithm is needed before the data are used by the estimator.

1.2.3 Task 3.0 Validation and Integration of Hybrid Model at Plant A

Key Responsibility: EPRI
EPRI will provide material data and inputs to initialize WVU's model. This starts with input data from Plant B, a well-instrumented operating combined cycle plant in the Southern Company fleet. Physics-based inputs (material degradation) will be integrated into a life management package, such as an upgraded version of EPRI's software, TULIP, or details on integrating with any tube life management approach.

Subtask 3.1 Project Management

Subtask 3.1 will cover all the project management concerns regarding the execution, coordination, and communication of this project. Deliverables to WVU, Southern Company, and DOE are expected every quarter in the form of succinct update/progress reports. Additionally, it is expected that semi-annual in-person project meetings will be needed to facilitate communication and planning, though these may take a virtual format as circumstances allow.

Subtask 3.2 Initialize the Model with Data from Plant B

This task is to collate the operational data from Plant B. Coordinate with Southern Company to provide to WVU the full set of operational data from Plant B as input data to the developing physics-based hybrid model. The principal deliverable for this subtask is the operational history from plant B over the relevant time span. This constitutes stress inputs (temperature changes) to the hybrid model. This subtask also covers the generation of additional materials data in support of the hybrid model including but not limited to microstructural features, cracking, oxide scale, etc. from relevant ex-service samples, including those form Plant B. Delivering the required inputs to WVU is work EPRI has completed. A more detailed evaluation of relevant damage mechanisms will enhance the physical understanding of the existing software tools. Oxide and ash can be significant insulators but can slough off in large chunks, increasing local temperature gradients (and stress). Oxide layers that crack can also act as precursor sites for damage accumulation locally. Development of the enhanced model to accept these additional factors is Subtask 3.5.

Subtask 3.3 Collect a Snapshot of Southern Company Host Site Operation

Separate from the data collection for initializing the database with the operational data from Plant B, additional data from the host site (Plant A) will be collected since that is a target for demonstrating the application value of this model. As such, the project will need access to their historical operation leading up to today. These data are needed to demonstrate to the model real-world startup cycles, load shed rates, thermal profiles, and more, all of which informs the stress conditions the components have seen and might see in the future. The deliverable from this task will be a simple package of to-be-determined operational data over a multi-year period. Collecting this data from the Plant A historian and sharing with WVU has been completed with additional data reports expected throughout the project.

Subtask 3.4 Pilot Demonstration of Model

It is important early in this project to identify a demonstration component of interest (COI) so that it can be additionally instrumented. Power plants are not known for large numbers of extra data collection points, but to properly validate the model predictions, the project may have to install additional sensors (e.g. thermocouples, strain gauges) in strategic locations. This requires working with the utility partner and host site to identify the COI (a superheater/reheater) and instrument it as needed by the hybrid model. This process needs to be started early to maximize the amount of data collected: a minimum of 12 months.

With a COI identified, a baseline condition inspection will be needed. This may or may not include material sampling, but at the very least optical and dimensional analysis are needed to compare to measurements taken at the end of this experiment. This may include material sampling of regions of interest for further characterization.

Subtask 3.5 Enhance Software

EPRI has functioning physics-based software tools that serve the power generation industry in their life assessment of boilers. These software packages take into consideration the interaction between pressure, temperature, geometry, material, and other inputs to derive a prediction of life consumed from long-term overheat (creep damage), ultimately leading to tube leaks which may force a plant shut down. Enhancements to the software will be focused on functional integration of the WVU hybrid model, materials damage models, and the dynamic ability to predict based on a continuous stream of inputs.

2 Development and validation of first-principles model and nonlinear state estimator using extended Kalman filter for differential algebraic equation (DAE) systems

2.1 Introduction

Fossil-fueled fired power plants still provide the majority of electricity worldwide, accounting for over 60% of the world's 25 trillion kWh of electricity production [1]. However, due to increased penetration of intermittent renewables to the energy grid, most power plants are forced to cycle loads frequently and rapidly, operate at low load conditions for a sustained period, and startup and shut down often over a hundred times a year [2–4]. These operating conditions may be significantly outside the system's original design parameters and risk accelerating damage to the boiler components, especially high-temperature superheater and reheater systems, due to unanticipated local fluctuations in temperature during load changes [5]. Superheater and reheater systems encountering high-temperature operations in pulverized coal (PC) and natural gas combined cycle (NGCC) power plants are vulnerable to failure caused by creep and fatigue damages due to temperature excursion and periodic thermal stresses during cycling operations [6,7]. Equipment failure will lead to unexpected plant outages, driving up operation and maintenance costs and affecting the overall production and profitability of the plant [8] . To mitigate these issues, developing advanced condition monitoring tools using dynamic modeling and estimation can be instrumental in understanding the impacts of load-following. These tools can eventually help these plants to make efficient preventive maintenance plans and avoid undesired forced outages for improved flexibility without compromising safety and reliability.

Condition monitoring using sensor measurement data is widely used for the assessment of rotary equipment items such as steam and combustion turbines in operating plants. However, monitoring in static components such as those found in boilers is less common due to the difficulty of detecting critical failures and damage in different boiler parts using measurement data [9] . One critical measurement for condition monitoring of the superheater/reheater is the temperature. However, there may be no or a limited number of sensors for measuring temperatures in zones of principal interest due to harsh and corrosive environments in high-temperature regions that are most vulnerable to failure. As a result, such location temperatures are inferred from measurements taken at a safer location, especially in superheater/reheater components. Recent advances in the sensor technology using infrared thermal imaging cameras and similar techniques have been utilized to measure temperature in superheaters/reheaters [10,11]. However, these sensors are used for gathering terminal points data of superheater/reheater systems, and accurately capturing data of the entire system requires many advanced sensors which drives up the operational costs. To address these challenges, the present work considers first-principles dynamic modeling and estimation of spatio-temporal profiles in superheater/reheater boiler components using limited temperature measurement data [12].

The complex dynamics of boiler components can be modeled accurately using differential-algebraic equation (DAE) models with ordinary differential equations (ODEs) representing the differential state variables of the system such as temperature, whereas algebraic equations represent correlations, boundary conditions, algebraic variables, etc. [13,14]. Although extensive work is available in the literature on dynamic modeling of boiler components like superheater/reheater, many of them are data-driven models. Data-driven models using artificial neural networks (ANN) and recurrent neural

networks (RNN) have been proposed for predicting transient thermal performance in superheater systems using steady-state and dynamic operating data of steam flow and pressure [15–19]. Data-driven modeling approaches have also been developed for digital-twin development for predicting metal temperature in coal-fired boiler units [20]. Fuzzy-logic-based data-driven models have been developed for the performance characterization of a fluidized-bed boiler superheater [21], and for predicting the superheated steam temperature [22]. Hybrid modeling techniques using physics-based and data-driven models have been used for predicting superheated steam temperature dynamics [23,24]. A hybrid modeling approach has also been used for superheater steam temperature control [25]. However, the works referenced above utilizing data-driven models have limited spatial resolution depending on the availability of measurement data. As the high temperature locations may not be where the sensors are available and as the high temperature locations can change due to change in the operating conditions, data-driven models may not be adequate for condition monitoring of superheater/reheaters. In addition, the data-driven models are highly specific to a plant and specific operating conditions and are not generalizable whereas physics-based models offer more flexibility.

To develop monitoring tools that can be used for a broad range of operating conditions, dynamic modeling using first-principles equations based on mass and energy balances has been utilized to model superheater/reheater systems in PC and NGCC power plants [26]. Linear lumped parameter models have been used to model superheated steam temperature dynamics using energy balances on steam and metal tube walls [27,28]. State-space models have also been proposed based on steady-state modeling using mass and energy balance equations for heat transfer from gas to metal and metal to fluid for the superheater system of a boiler [29,30]. Although useful for specific monitoring applications, the linear lumped parameter and state-space model are inadequate for health monitoring of boiler components because the spatial variation of state variables is not represented by these models. During rapid changes in the firing, there can be considerable spatio-temporal evolution in temperature that will be desired to be modeled for condition monitoring [31]. Further, boiler dynamics are highly nonlinear thus leading to inaccuracies when linear models are used.

To obtain information about the spatial variation of process variables in a boiler, Boje and Eitelberg have proposed a distributed partial differential equation (PDE) model of a superheated steam header system in a power plant boiler [32]. Patel and Boje have modeled the convection pass of a once-through boiler using spatial discretization of each tube pass for steam side and pipe wall dynamics with additional consideration of flue gas side dynamics [33]. In the existing literature, there are numerous works using detailed distributed dynamic first-principles model for the superheater/reheater system which have been validated with industrial operating data [34–37] . However, most of these works have been validated using steady-state industrial data or limited dynamic data, during which only one input parameter has been modified at a time. A comprehensive modeling and simulation tool that includes detailed models of steam, flue gas, and metal tube dynamics has been developed for steam-generating process systems such as industrial boilers [38] and heat recovery steam generator (HRSG) systems [39]. Zima [40–42] has proposed distributed dynamic models for studying transients in superheater systems. These models have been applied to analyze how an industrial boiler system behaves during startup, shutdown, and acceptance tests. Although first-principles, distributed dynamic models of the superheater/reheater have been developed in the existing literature, these models often consider simple configurations that are not representative of configurations in typical industrial superheater/reheaters such as those in boilers and steam generators for power generation. Further, the models are validated with simulated data, or steady-state industrial data, or very limited dynamic industrial data such as from a step test. For condition

monitoring, as the damaging temperature transients often appear under rapid change in the plant load and at part-load conditions, it is desired that the model results are validated by comparing with industrial data collected under varying load and considerable dynamic operation of the plant. As the tube temperatures are the key variables that affect the thermal stress and tube life [43], it is desired to have good accuracy for the tube temperature results. Thus, it is desired to compare the tube temperature profile obtained from the model with the industrial data. However, comparison of the model results for the dynamic tube temperature profile with the industrial data has rarely been done in the open literature. It is also desired that the distributed process models of superheaters/reheaters are developed in a modular manner so that they can be readily used for representing vastly different configurations of industrial superheaters/reheaters.

As both the process model and industrial measurements are expected to have errors, an optimal estimation approach can be instrumental to synergistically consider both process model and measurements and yield more accurate estimates of both measured and unmeasured variables [44]. State and parameter estimation of boiler components has been studied using the Kalman filter (KF), its variants, or optimization-based moving horizon estimation methods [45]. The traditional KF can lead to inaccuracies for nonlinear systems. For nonlinear systems, one of the approximate approaches that is not as computationally intensive as the fully nonlinear filters is the extended Kalman filter (EKF). Linear adaptive KF using augmented state equations and continuous-discrete KF has been used for state estimation of industrial superheater systems [28,32,33]. A linear KF using linear lumped models [27], a static state estimator using a steady-state model [46] and a moving horizon estimator [47] have also been used for estimating main steam and final superheater tube temperatures for boiler systems. However,due to the high nonlinearity of superheater/reheater systems, linear Kalman filters can lead to inaccuracies thus nonlinear filtering algorithm such as EKF or unscented Kalman filter (UKF) are desired.

EKF based estimation approaches have been used by Nair et al. [48] for estimating furnace gas parameters, primarily specific enthalpy and density, and by Jonsson and Palsson [49–51] for estimating parameters used for calculating the heat transfer coefficient. An EKF based approach have also been used for estimating fouling and leak detection in superheater/reheater systems [52–55]. UKF based estimation techniques have also been used for state estimation in an industrial steam superheater system [35], a boiler turbine unit [56], and a circulating fluidized bed boiler [57]. Recently, a modified UKF estimator for studying hot corrosion in waterwall section of industrial steam boilers has been implemented and validated by using the industrila data [58]. In summary, EKFs or UKFs have been mainly used for data-driven or lumped parameter models given by ODEs or PDEs. There is a lack of work in the open literature on state estimation of boiler systems represented by the DAE model, which is essential to completely characterize the complex dynamics of working fluids and tube walls in these equipment items. In the existing literature on the EKF for non-linear DAEs, the algebraic equations are typically considered noise-free. Thus, the algebraic equations can be differentiated, converting the system to an implicit ODE [59–61]. For some systems, there can be both exact and uncertain algebraic equations. For uncertain algebraic equations, the white noise term cannot be differentiated since the derivative of white noise is not well-defined. A modified EKF that can handle uncertainties in differential and algebraic equations and exact algebraic equations, and uncertain measurement noise characteristics is needed [62]. In this work a modified EKF algorithm for DAE system and the distributed 2-D DAE model has been utilized to estimate critical and unmeasured variables for the superheater [63].

2.2 Dynamic Distributed DAE Model of the Superheater/Reheater

The dynamic 2-D distributed differential algebraic equation (DAE) model for the superheater/reheater section is developed based on the fundamental mass and energy balance equations. The model is developed using differential equations for the rate of change of mass and energy for each flowing stream. The algebraic equations calculate the heat transfer coefficient and Fig. 2.1 represents a module of the superheater/reheater with crossflow configuration that is used in parallel and series combinations to represent varieties of superheater/reheater geometry. The direction and orientation of each flow stream, i.e., vertical or horizontal, is generic and can be modified based on the specific configuration. The 1-D mass and energy balances are solved for each stream relative to its assigned flow direction making the model overall 2D. Convective heat transfer is the dominant mechanism under a thin-wall assumption.

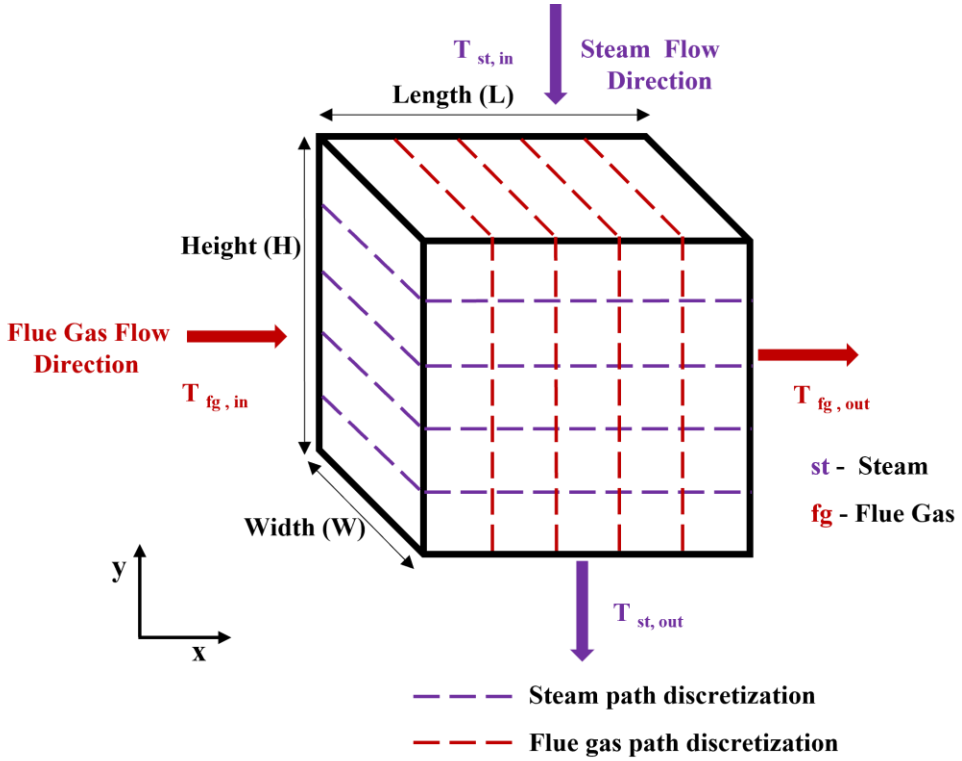


Fig. 2.1. Module representation of the superheater/reheater distributed model

Mass and energy balances for both the steam and flue gas flow are given by [64] :
For steam side:

$$\frac{\partial \rho_{st}}{\partial t} + \frac{\partial (\rho_{st} v_{st})}{\partial y} = 0 \tag{2.1}$$

$$\frac{\partial T_{st}}{\partial t} + v_{st} \frac{\partial T_{st}}{\partial y} = + \frac{q_{wall,st}}{\rho_{st}\, c_{p,st}} \tag{2.2}$$

For flue gas side:

$$\frac{\partial \rho_{fg}}{\partial t} + \frac{\partial (\rho_{fg} v_{fg})}{\partial x} = 0 \tag{2.3}$$

$$\frac{\partial T_{fg}}{\partial t} + v_{fg} \frac{\partial T_{fg}}{\partial x} = - \frac{q_{wall,fg}}{\rho_{fg}\, c_{p,fg}} \tag{2.4}$$

where ρ and v are steam and flue gas density and velocity, T is steam and flue gas temperature, c_p is the specific heat capacity of steam and flue gas for each control volume depicted in Fig. 2.1.

In Eq. (2.2) and (2.4), q_{wall} is the volumetric heat flux of the steam and flue gas side, which is + ve for the steam side (gain of heat) and - ve for the flue gas side (loss of heat).

The bulk velocities for steam and flue gas calculated based on inlet mass, geometry, and density for both streams are as follows:

$$v_{st} = \frac{m_{st}}{N_t \frac{\pi}{4} D_{i,t}^2 \rho_{st}} \tag{2.5}$$

$$v_{fg} = \frac{m_{fg}}{L_{fg} W_{fg} \rho_{fg}} \tag{2.6}$$

where v_{st} and v_{fg} are steam and flue gas bulk velocities, m_{st} and m_{fg} are steam and flue gas inlet mass flowrates, ρ_{st} and ρ_{fg} are steam and flue gas densities, N_t is the number of tubes in the system, $D_{i,t}$ is the inside diameter of steam tubes, L_{fg} and W_{fg} are the length and width of the flue gas zone.

The tube side dynamics are given by the equation [65]:

$$m_t c_{p_t} \frac{dT_t}{dt} = h_{conv,fg} A_0 (T_{fg,in} - T_t) - h_{conv,st} A_i (T_t - T_{st,out}) \tag{2.7}$$

where m_t is the metal mass of the tubes, c_{p_t} is the specific heat capacity of tube material, $T_{fg,in}$ is the flue gas inlet temperature, $T_{st,out}$ is the steam outlet temperature, $h_{conv,fg}$ and $h_{conv,st}$ are the heat transfer coefficients on the flue gas and steam side, A_0 is the total outside heat transfer area, A_i is the total inside heat transfer area and T_t is the tube wall temperature.

The convective heat flux on the steam side considering the heat transfer from the tube wall to the steam side is given by [66]:

$$q_{st} = h_{conv,st} (T_t - T_{st,out}) \tag{2.8}$$

Similarly, the convective heat flux on the flue gas side considering the heat transfer from the flue gas side to the outside of the tube wall is given by [66]:

$$q_{fg} = h_{conv,fg}\,(T_{fg,in} - T_t) \tag{2.9}$$

where q_{st} and q_{fg} are the heat flux and $h_{conv,st}$ and $h_{conv,fg}$ are the heat transfer coefficients for steam and flue gas for each control volume.

The volumetric heat flux of steam and flue gas side, q_{wall} , used in Eq. (2.2) and (2.4), is calculated as:

$$q_{wall,st} = q_{st}\frac{\pi\,\mathrm{D}_{i,t}L_tN_t}{\frac{\pi}{4}\,\mathrm{D}_{i,t}^2L_tN_t} = q_{st}\frac{\pi}{D_{it}} \tag{2.10}$$

$$q_{wall,fg} = q_{fg}\frac{A_{o,t}}{V_{fg}} = q_{fg}\frac{\pi\,\mathrm{D}_{o,t}L_tN_t}{L_{fg}H_{fg}W_{fg}} \tag{2.11}$$

The steam side heat transfer coefficient, $h_{conv,st}$, is calculated using Gnielinski correlation for flow through smooth tubes as [66]:

$$Nu_{st} = \frac{\left(\frac{\xi_{st}}{8}\right)Re_{st}Pr_{st}}{1 + 12.7\sqrt{\frac{\xi_{st}}{8}}\left(Pr_{st}^{\frac{2}{3}} - 1\right)}\left(1 + \frac{D_{i,t}}{L_t}\right)^{\frac{2}{3}} \tag{2.12}$$

where L_t is the length of the tube considered and the parameter , ξ_{st} is given as:

$$\xi_{st} = (1.8\log_{10}Re_{st} - 1.5)^{-2} \tag{2.13}$$

The Nusselt number in Eq. (2.12) is calculated as:

$$Nu_{st} = \frac{h_{conv,st}L_{char,st}}{k_{st}} \tag{2.14}$$

where $L_{char,st}$ is the characteristic length given as:

$$L_{char,st} = D_{i,t} \tag{2.15}$$

The other two terms Re_{st} and Pr_{st} in Eq. (2.12) are the Reynolds and Prandtl number calculated as:

$$Re_{st} = \frac{\rho_{st}v_{st}L_{char,st}}{\mu_{st}} \tag{2.16}$$

$$Pr_{st} = \frac{\mu_{st}c_{p,st}}{k_{st}} \tag{2.17}$$

where μ_{st} is the dynamic viscosity, k_{st} is the thermal conductivity and $c_{p,st}$ is the specific heat capacity of the steam. All of these properties are calculated using the functions defined in the IAPWS R7-97 formulation for the properties of water/steam [67].

The flue gas heat transfer coefficient, $h_{conv,fg}$, is calculated by Gnielinski correlation for flow across a tube bundle where factors are calculated based on the transverse and longitudinal pitch (s_1 and s_2) as well as configuration [66]:

$$a_t = \frac{s_{1,t}}{D_{o,t}} \tag{2.18}$$

$$b_t = \frac{s_{2,t}}{D_{o,t}} \tag{2.19}$$

$$\text{and} \quad \psi_t = 1 - \frac{\pi}{4a_t} \quad \text{for } b_t \geq 1 \tag{2.20}$$

where ψ_t is the void fraction and $D_{o,t}$ is the outer tube diameter.
Using the above-defined factors from Eq. (2.18) to (2.20) an arrangement factor, f_A, is calculated based on the arrangement of the tubes (i.e., inline or staggered) as follows:

$$f_{A,il,fg} = 1 + \frac{0.7\left(\frac{b_t}{a_t} - 0.3\right)}{\psi_t^{1.5}\left(\frac{b_t}{a_t} + 0.7\right)^2} \tag{2.21}$$

$$f_{A,stag,fg} = 1 + \frac{2}{3b_t} \tag{2.22}$$

Reynolds and Nusselt numbers for the flue gas side are calculated as:

$$Re_{fg} = \frac{\rho_{fg} v_{fg} L_{char,fg}}{\mu_{fg} \psi_t} \tag{2.23}$$

$$Nu_{bundle,fg} = \frac{h_{conv,fg} L_{char,fg}}{k_{fg}} \tag{2.24}$$

where $L_{char,fg}$ is defined as the characteristic length as follows:

$$L_{char,fg} = \left(\frac{\pi}{2}\right) D_{o,t} \tag{2.25}$$

Prandtl number is given as:

$$Pr_{fg} = \frac{\mu_{fg} c_{p,fg}}{k_{fg}} \tag{2.26}$$

where μ_{fg} is the dynamic viscosity, k_{fg} is the thermal conductivity and $c_{p,fg}$ is the specific heat capacity of the flue gas.
Furthermore, Nusselt number defined in Eq. (2.24) is calculated based on laminar and turbulent components as follows:

$$Nu_{lam,fg} = 0.664\, Re_{fg}^{\frac{1}{2}}\, Pr_{fg}^{\frac{1}{3}} \tag{2.27}$$

$$Nu_{turb,fg} = \frac{0.037\, Re_{fg}^{0.8} Pr_{fg}}{1 + 2.443\, Re_{fg}^{-.01}\left(Pr_{fg}^{\frac{2}{3}} - 1\right)} \tag{2.28}$$

$$Nu_{fg} = 0.3 + \sqrt{Nu_{lam,fg}^2 + Nu_{turb,fg}^2} \tag{2.29}$$

$$Nu_{bundle,fg} = f_{A,fg} Nu_{fg} \tag{2.30}$$

2.3 Estimation algorithm: Modified differential algebraic equation-extended Kalman filtering (DAE-EKF)

The distributed 2-D DAE model developed for the boiler systems in Section 2.2, as shown in Fig. 2.1, is a nonlinear DAE system. A modification to the existing EKF algorithm available in literature [68] has been undertaken to include noise in both the differential and algebraic equations and consider algebraic equations to be both exact and uncertain.

The mathematical formulation of the algorithm assuming a continuous non-linear dynamic DAE system with discrete measurements available is defined as:

$$x_k = x_{k-1} + \int_{(k-1)\Delta t}^{k\Delta t} f\big(x(t), r(t)\big)dt + \omega_k \tag{2.31}$$

$$0 = g(x_k, r_k) + \sigma_k \tag{2.32}$$

$$\text{and } y_k = z(x_k, r_k) + \varepsilon_k \tag{2.33}$$

where x, r, y are the differential state variables, algebraic state variables and measurements, respectively. The variables ω and σ are the differential and algebraic state process noise, respectively, and ε is the measurement noise. The process (ω, σ) and measurement noise (ε) are considered as zero-mean white Gaussian noise given as $\omega \sim N(0, Q)$, $\sigma \sim N(0, S)$ and $\varepsilon \sim N(0, R)$. Here, 'Q', 'S', and 'R' are defined as the process (differential and algebraic) and measurement noise covariances, acting as the filters' tuning parameters. The application of this algorithm follows a two-step approach of prediction and correction steps to propagate the system states forward in time.

Initialization: Starting with initial values $\hat{x}_{k-1|k-1} = E[x(t_{k-1})]$
and $M_{k-1|k-1}$ (error covariance of state variables)

Prediction Step: Time Update

- The states are projected ahead in time using Eq. (2.31) and (2.32) without the ω_k and σ_k terms and solved using any commercial DAE solver (like MATLAB '*ode15s*').

$$\hat{x}_{k|k-1} = \hat{x}_{k-1|k-1} + \int_{(k-1)\Delta t}^{k\Delta t} f\big(x(t), r(t)\big)dt \tag{2.34}$$

$$0 = g\big(\hat{x}_{k|k-1}, \hat{r}_{k|k-1}\big) \tag{2.35}$$

- The error covariance of the states for the DAE is not possible to compute by augmenting the state together (differential and algebraic state) as proposed in the work of Mandela et al. [59] since the white noise term σ_k in the algebraic Eq. (2.32) cannot be differentiated to convert the DAE system into an implicit ODE system. Hence, in order to propagate the error covariance matrix forward in time, we first need to linearize the DAE system described in Eq. (2.31) and (2.32) around $\hat{x}_{k-1|k-1}$ and $\hat{r}_{k-1|k-1}$ as follows:

$$\dot{x} = Ex + Fr \tag{2.36}$$

$$0 = Gx + Hr \tag{2.37}$$

where *E*, *F*, *G*, and *H* comprise the Jacobian matrix of the DAE model, with each state given as:

$$J = \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} & \dfrac{\partial f}{\partial r} \\ \dfrac{\partial g}{\partial x} & \dfrac{\partial g}{\partial r} \end{bmatrix} \tag{2.38}$$

Rearranging Eq. (2.36) and (2.37) to obtain that equation in terms of differential variable, *x*, we get:

$$r = (-H^{-1}G\,)x \tag{2.39}$$

$$\dot{x} = (E - FH^{-1}G)x \tag{2.40}$$

The error covariance matrix is now calculated by splitting it into separate blocks for differential and algebraic states as proposed in the approach of Mobed et al. [62] as follows:

$$M_{k|k-1} = \begin{bmatrix} M^{xx}_{k|k-1} & M^{xr}_{k|k-1} \\ M^{rx}_{k|k-1} & M^{rr}_{k|k-1} \end{bmatrix} \tag{2.41}$$

The individual terms of Eq. (2.41) are defined as follows:

$$M^{xx}_{k|k-1} = \tau\, M^{xx}_{k-1|k-1}\tau^T + Q_k \tag{2.42}$$

where $M^{xx}_{k|k-1}$ is the error covariance of the differential states with $M^{xx}_{k-1|k-1} = M_{k-1|k-1}$ (initial covariance assumed at t =0) and τ is defined as a transition matrix for differential states given as:

$$\tau = e^{(E-FH^{-1}G)\Delta t} \tag{2.43}$$

The other terms in Eq. (2.41) are as follows:

$$M_{k|k-1}^{rr} = (H^{-1}G)M_{k|k-1}^{xx}(H^{-1}G)^T + H^{-1}S_k{H^{-1}}^T \tag{2.44}$$

$$M_{k|k-1}^{xr} = M_{k|k-1}^{xx}(H^{-1}G)^T \tag{2.45}$$

$$M_{k|k-1}^{rx} = (H^{-1}G)M_{k|k-1}^{xx} \tag{2.46}$$

where $M_{k|k-1}^{rr}$ is the error covariance of the algebraic states, $M_{k|k-1}^{xr}$ and $M_{k|k-1}^{rx}$ are the cross-covariance of the differential and algebraic states with each other assuming that the algebraic states are the linear transformation of differential states as shown in Eq.(2.39).

Correction Step: Measurement Update

- The Kalman gain of the system is calculated as:

$$K_k = M_{k|k-1}{Z_k^{aug}}^T \left(Z_k^{aug} M_{k|k-1} {Z^{aug}}_k^T + R_k\right)^{-1} \tag{2.47}$$

  where, $Z_k^{aug} = [\frac{\partial z}{\partial x} \frac{\partial z}{\partial r}]$ (Jacobian matrix of measurement model with each differential and algebraic state).

- The state estimate is updated using the measurement, y_k, as follows:

$$\hat{x}^{aug}{}_{k|k} = \hat{x}^{aug}{}_{k|k-1} + K_k(y_k - z(\hat{x}^{aug}{}_{k|k-1})) \tag{2.48}$$

  where $\hat{x}^{aug}{}_{k|k-1}$ is the augmented state estimate (combined differential and algebraic states).

- Now, from Eq. (2.48), out of the updated augmented states $\hat{x}^{aug}{}_{k|k}$, only the updated estimates of differential states are retained. The updated algebraic states estimates are calculated by using the algebraic equation as follows:

$$0 = g\left(\hat{x}_{k|k-1}, \hat{r}_{k|k-1}\right) \tag{2.49}$$

- The updated error covariance is given as follows:

$$M_{k|k} = (I - K_k Z_k^{aug}) M_{k|k-1} \tag{2.50}$$

2.4 Case Study: Final Superheater System (Heat Recovery Steam Generator- NGCC Plant B)

The validity of the superheater model and the estimator has been assessed by considering the performance of the final superheater in the heat recovery steam generator (HRSG) section of a 600 MW NGCC plant. After passing through the superheater system, the steam is collected in an outlet header before passing through a steam line to a steam turbine. The steam in the main steam line is at nominal conditions of 535 °C and 10 MPa. The configuration of the superheater section is shown schematically in Fig. 2.2**.**

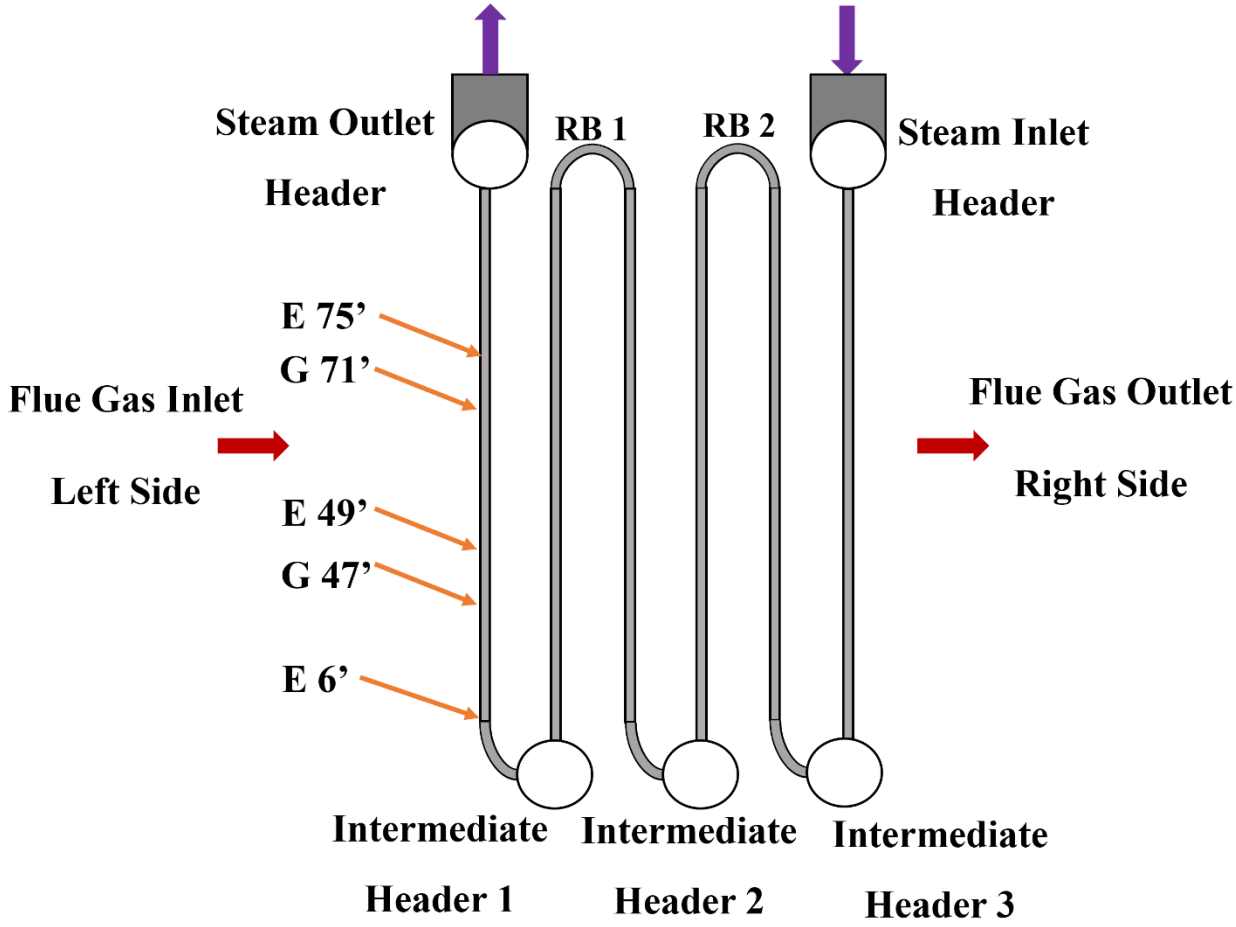


Fig. 2.2. Schematic representation of the final superheater system used in this study – locations identified on the left are at the elevation (in ft) indicated. Steam exiting the outlet header is transmitted through a main steam pipe to a steam turbine.

As shown in Fig. 2.2, the selected system is a six-pass final superheater system of an HRSG unit. The steam from the outlet of this superheater system goes through a main steam pipe to the high pressure (HP) steam turbine. The given system consists of 74 identical flow circuits, each with six-pass tubes arranged behind each other. The exhaust gas stream from the combustion turbine flows from left to right in the horizontal direction. In the given system, the steam is assumed to flow unmixed as a plug flow, while the flue gas is assumed to be well mixed. The specific industrial system undertaken for consideration was well instrumented, with many more thermocouples being available than in typical superheater system plants. The main steam temperature was measured, as

this is one of the most critical and tightly controlled variables. Outside tube wall temperatures were measured using mounted thermocouples for this system at locations E 75', E 49', E 6', return bend 1 (RB 1), and return bend 2 (RB 2), as shown in Fig. 2.2. Also, flue gas temperatures along the flow path were measured using thermocouples in the flue gas path at locations G 71' and G 47' as shown in Fig. 2.2**.**

Fig. 2.3 shows the configuration and control volumes of the six-pass final superheater system. The tubes in the system were in a staggered arrangement. The PDEs for the superheater system described in Section 2.2 have been converted to ODEs by using method of lines leading to a system of ordinary differential-algebraic equations (ODAEs). Finally, all the ODAEs are solved together in MATLAB by using DAE solver '*ode15s*'.

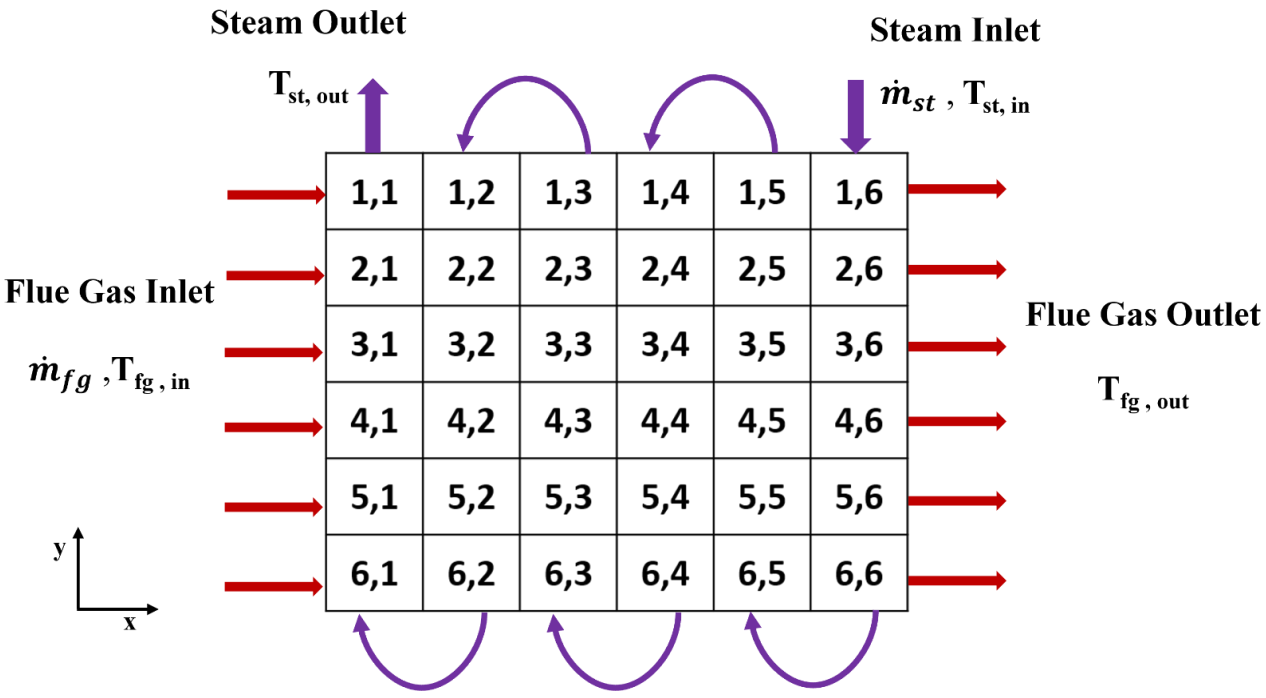


Fig. 2.3. Representation of the six-pass final superheater system using control volumes (the second index number in each control volume refers to the tube pass number for the superheater system).

2.5 Results and Discussion

2.5.1 Model and Estimator Validation

Extensive operational datasets were made available from the industrial partner plant (IPP). From the overall information, a dataset covering 15 h was considered for validation. Fig. 2.4 shows plots of input variables taken from the industrial data for the final superheater system. The plot shows that the data contains sufficient variabilities with frequent cycling between varying load conditions. Two distinct near-steady zones are identified from the dataset, from about 0-5 h and 10-15 h.

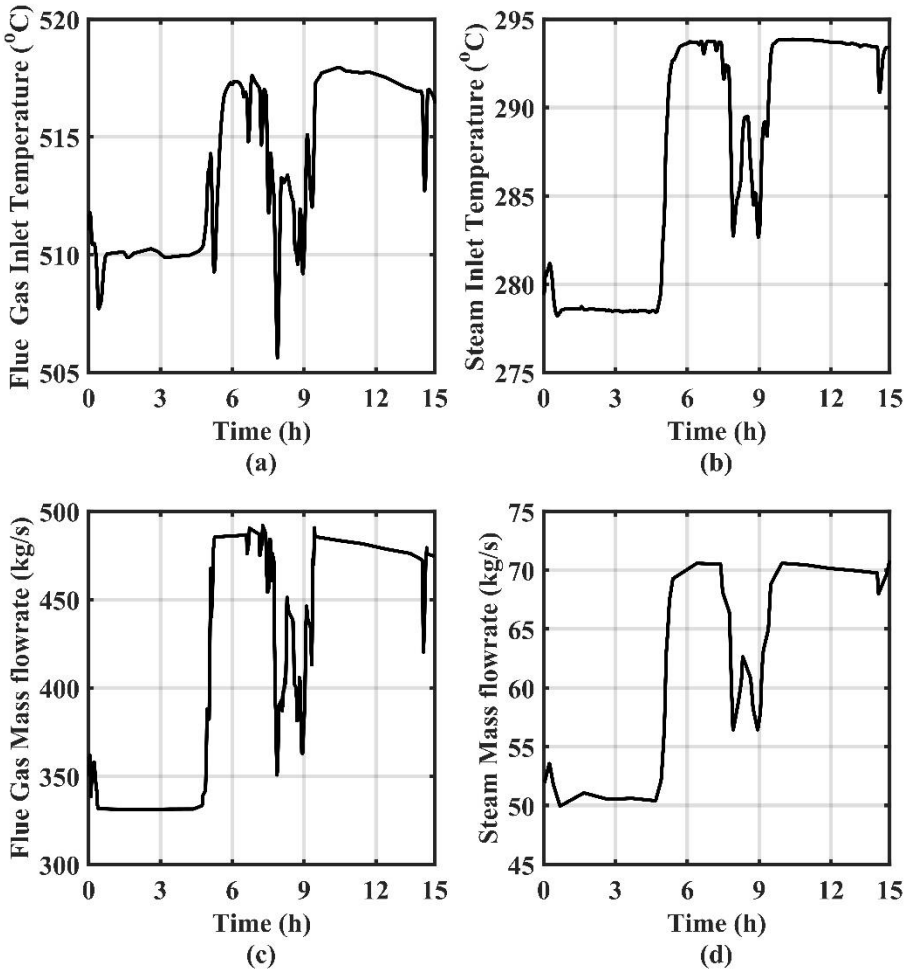


Fig. 2.4. Input variables considered for the industrial data. (a) Flue gas inlet temperature ($T_{fg,in}$), (b) Steam inlet temperature ($T_{st,in}$), (c) Flue gas mass flowrate ($\dot{m}_{fg}$), (d) Steam mass flowrate ($\dot{m}_{st}$), (as shown in Fig. 2.3).

In all the figures shown subsequently from Fig. 2.5 to Fig. 2.8 and in **Appendix A**, the legend "Model Results" refers to the predictions obtained by solving the first-principles DAE model described in Section 2.2 for the superheater system without incorporating the estimator algorithm described in Section 2.3, and the legend "Estimator Results" were the corrected prediction obtained from the estimator algorithm which also uses the DAE model and measured data from the plant. Fig. 2.5 compares the measured data of the main steam outlet temperature with predictions made using the model (red line) and the estimator result (black line). It is observed that while the uncorrected model results compare well with the measured data during steady-state operation between 0 and 5 h and again between 10 and 15 h, high differences between the data and prediction can be found during fast transients particularly around 6 h and 8 h. However, the estimator results matched well with the industrial data with an RMSE of 0.2 ^{o}C observed for the entire duration.

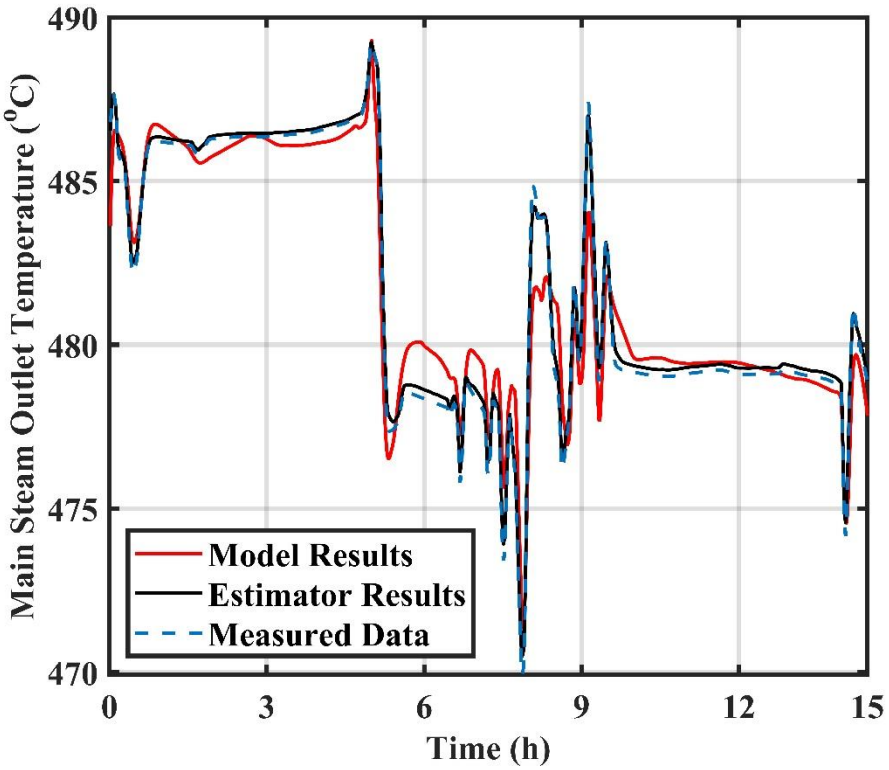


Fig. 2.5. Comparison of the measured data of the main steam outlet temperature with the uncorrected-model-prediction and estimator-corrected results.

Fig. 2.6 and Fig. 2.7 compare the measured data of flue gas temperature at location G 47' (see Fig. 2.2**)** and tube wall temperature at location E 75' (see Fig. 2.2**),** respectively**,** with the model and estimator results. Both these measurements are located at the upstream side of the tube (i.e., where the flue gas enters the superheater). Also, these measurements were only available for some tubes of the total 74 flow circuits located along the width of the superheater system. Hence, for validation studies, the average of these measurements was considered for comparison. Also, the maximum and minimum measured temperatures are plotted showing the span of the temperature readings. Fig. 2.6 and Fig. 2.7 show that the model results have deviation from the average measured data not only during the transient operation but also during the two near-steady operating regions. However, the model results are within the bounds of maximum and minimum measurement values for both variables. The estimator results are found to be very close to the

average measured values for both variables in the entire operating time. Validation plots of other tube wall and flue gas temperatures measured locations (see Fig. 2.2 **),** can be found in **Appendix A**.

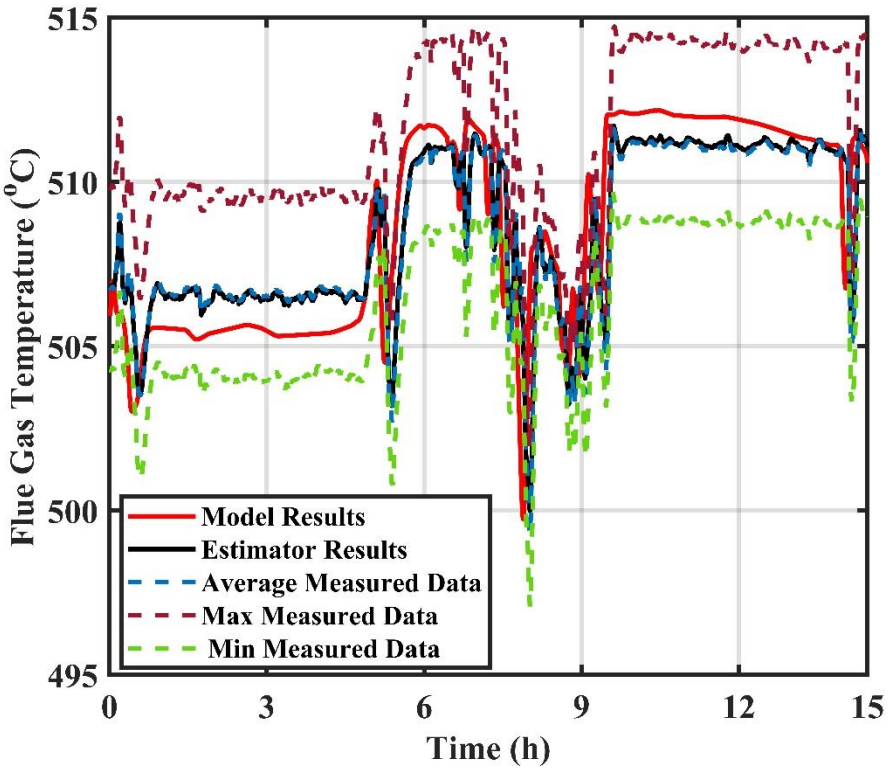


Fig. 2.6. Comparison of the model and estimator results with the measured data for the flue gas temperature at an elevation of 47 feet (G 47').

The root mean square error (RMSE) of the model and estimator results with respect to the measurement data for all three variables is reported in **Table 2.1**, which shows that the model error was reduced significantly for all three variables when the estimator was used.

Table 2.1: RMSE for three measured variables for the final superheater system (NGCC plant)

Root Mean Square Error (RMSE)		
Variable name	**Model RMSE error (ºC) (Measured-Model)**	**Estimator RMSE error (ºC) (Measured-Estimated)**
Main steam outlet temperature ($T_{st,\ out}$)	1	0.2
Flue Gas Temperature (G 47')	1.8	0.17
Tube Temperature (E 75')	2.5	0.28

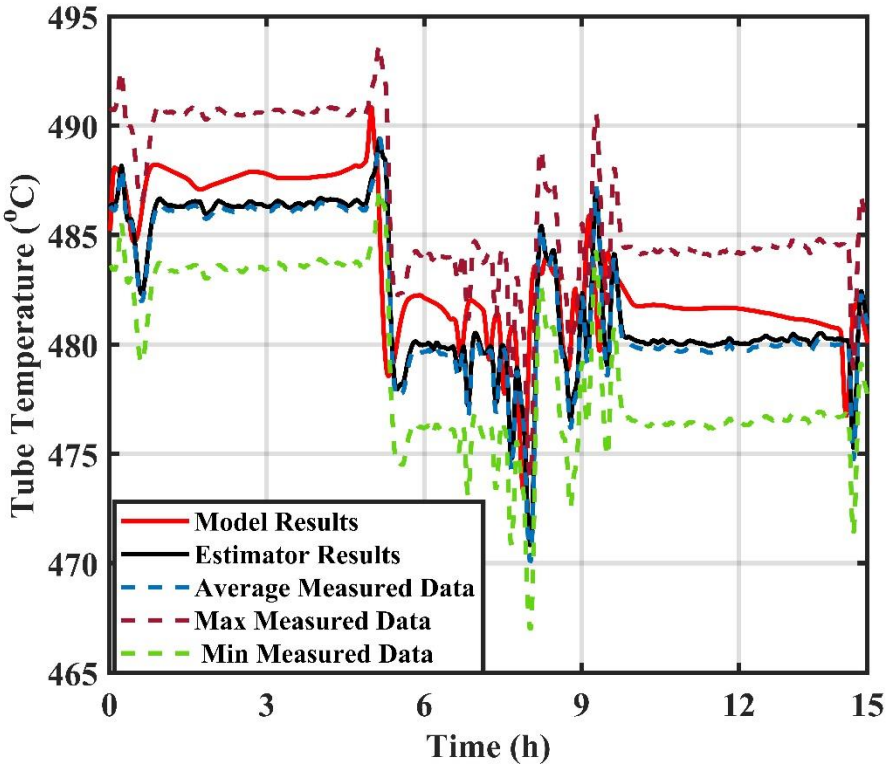


Fig. 2.7. Comparison of the model and estimator results with the measured data for the tube temperature at an elevation of 75 feet (E 75').

One key advantage of using the first principles model and the estimator framework is the estimation of unmeasured variables. An estimator's major utility is that with some measured variables for any system, such as the main steam temperature in the superheater system, estimates of other unmeasured variables can be improved. As an example, Fig. 2.8 shows the model and estimator results of the flue gas outlet temperature exiting from the right side (downstream side) of the final superheater (see Fig. 2.2**)**. This variable is difficult to measure by placing a sensor due to harsh conditions on the flue gas side where the sensor cannot survive for long. Estimation results of other unmeasured variables of interest, such as the temperature distribution in intermediate headers for outer tube walls and flue gas temperatures (see Fig. 2.2**),** can be found in **Appendix A**.

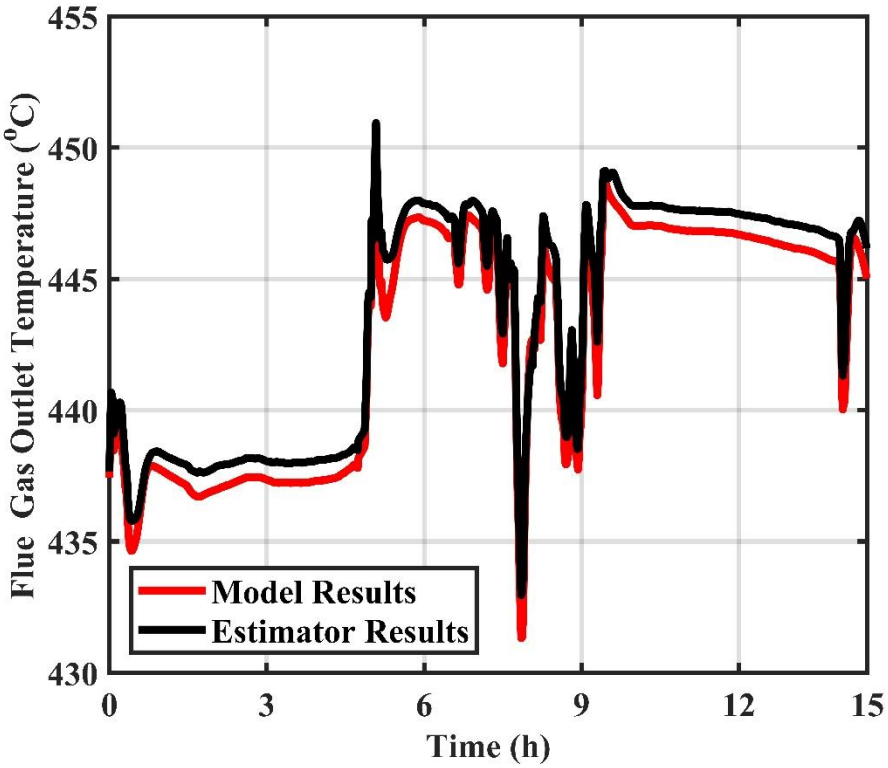


Fig. 2.8. Estimation of the unmeasured flue gas outlet temperature ($T_{fg,out}$, as shown in Fig. 2.3).

2.5.2 Sensitivity Studies

Studies were conducted to evaluate the estimator performance for a small dataset that captures the load-following operation by the power plant, i.e., the plant goes from the lowest to the highest load, stays there for a while, and then returns to the low load condition. In addition, it is desired to evaluate the estimator performance when there is a large discrepancy between the model and the plant, as well as when there is high measurement noise. The natural gas (NG) fuel feed flowrate profile and the transient of the input/boundary variables (steam and flue gas inlet temperatures and flowrates to the superheater) considered for the study are shown in Fig. 2.9 and **Fig. 2.10**, respectively.

The natural gas fuel feed flowrate shown in Fig. 2.9 has been normalized using "max-min normalization" to protect the proprietary industrial data. The natural gas feed profile shown in Fig. 2.9 varies from a normalized lower value of zero (corresponding to the low load condition of 350 MW) to a normalized upper value of one (corresponding to the high load condition of 600 MW). Furthermore, random fluctuations are also considered in the profile between 0.5-1 h , where the feed flow rate drops and then increases again shortly thereafter. These random changes can be attributed to unknown phenomena affecting the superheater system, such as the gas turbine firing mechanism or the airflow rate in the turbine, which are not captured in the first principles model described in Section 2.2.

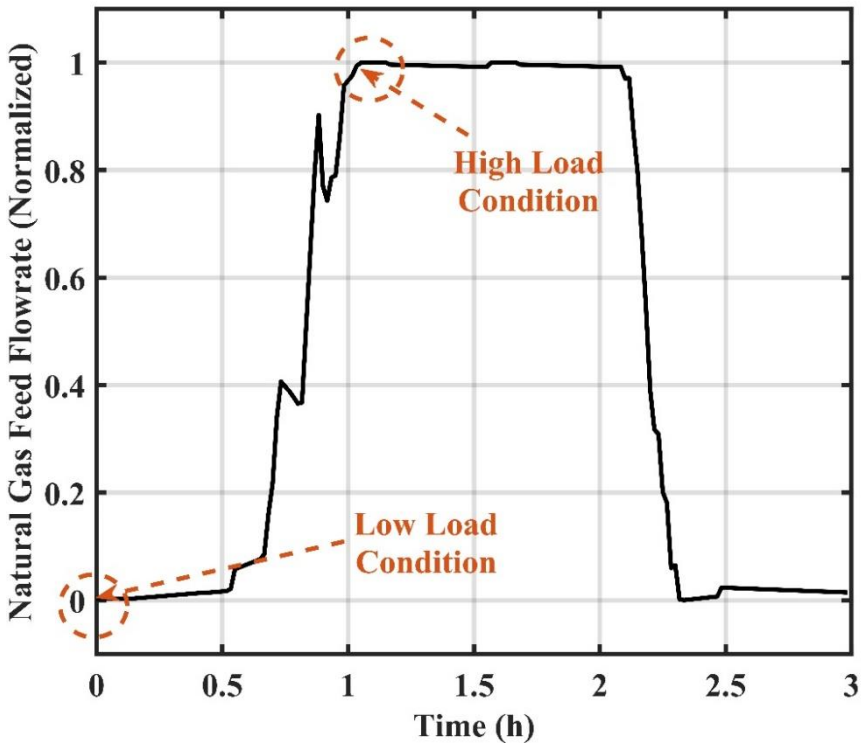


Fig. 2.9. Natural gas feed profile for the sensitivity study.

While the profiles of input variables shown in **Fig. 2.10** are as expected due to the load-following operation, some additional dynamics are considered for the flue gas side mass flowrate and inlet temperature – specifically an inverse profile in the temperature at 1 h and at 2.5 h. As seen in Fig. 2.9, these changes are arbitrary and are mainly considered to evaluate the estimator performance under rapid changes of some unknown yet key variables that significantly affect the superheater profile but are not captured in the DAE model. The noisy measurement data are obtained by adding normally distributed Gaussian white noise using MATLAB *randn* function with a standard deviation of 0.5, 0.8, and 0.5 from the actual values for outlet steam temperature, outlet flue gas temperature, and tube temperatures for all the grid points as shown in Fig. 2.3, respectively. The standard deviation considered for flue gas temperature is higher than the steam and tube temperatures because random fluctuations on the flue gas side lead to higher discrepancy while measuring the flue gas temperatures. To study the estimator performance, a higher mismatch between the model and the truth is generated by changing the constant coefficient parameter of Eq. (2.29) for the flue gas side Nusselt number.

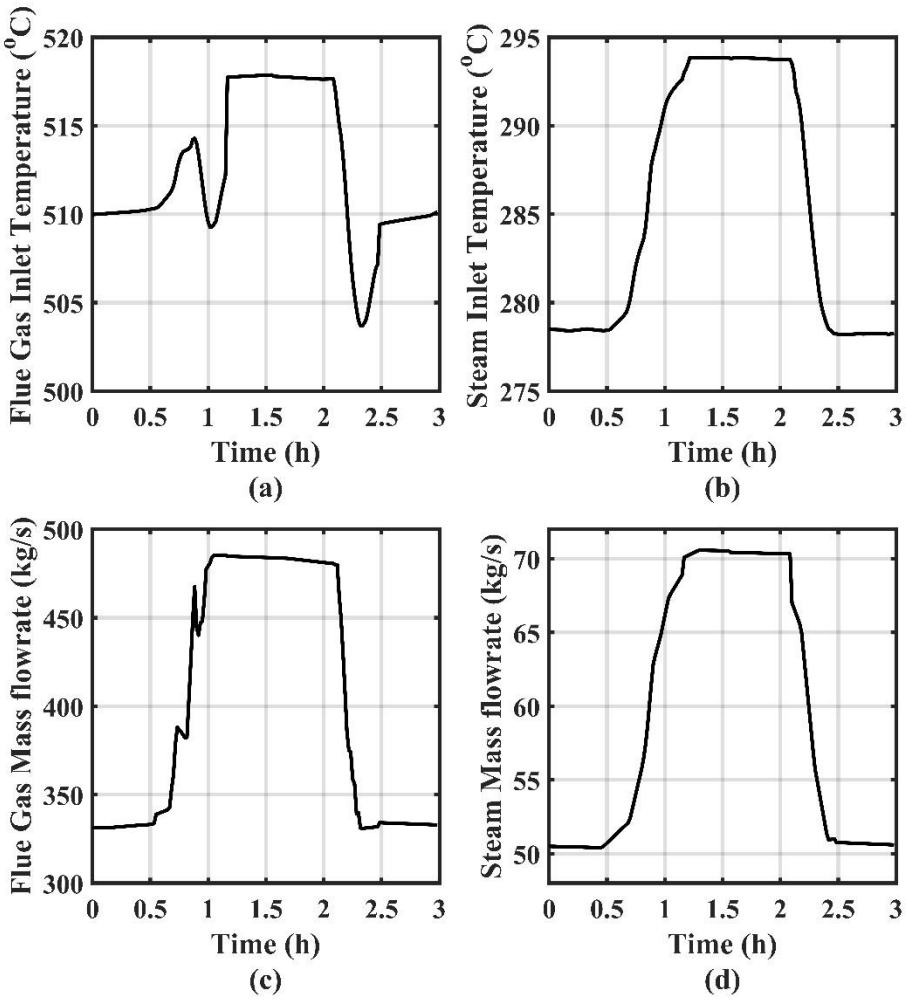


Fig. 2.10. Profiles of boundary variables for the sensitivity study. (a) Flue gas inlet temperature ($T_{fg,in}$), (b) Steam inlet temperature ($T_{st,in}$), (c) Flue gas mass flowrate ($\dot{m}_{fg}$),(d) Steam mass flowrate ($\dot{m}_{st}$), (as shown in Fig. 2.3).

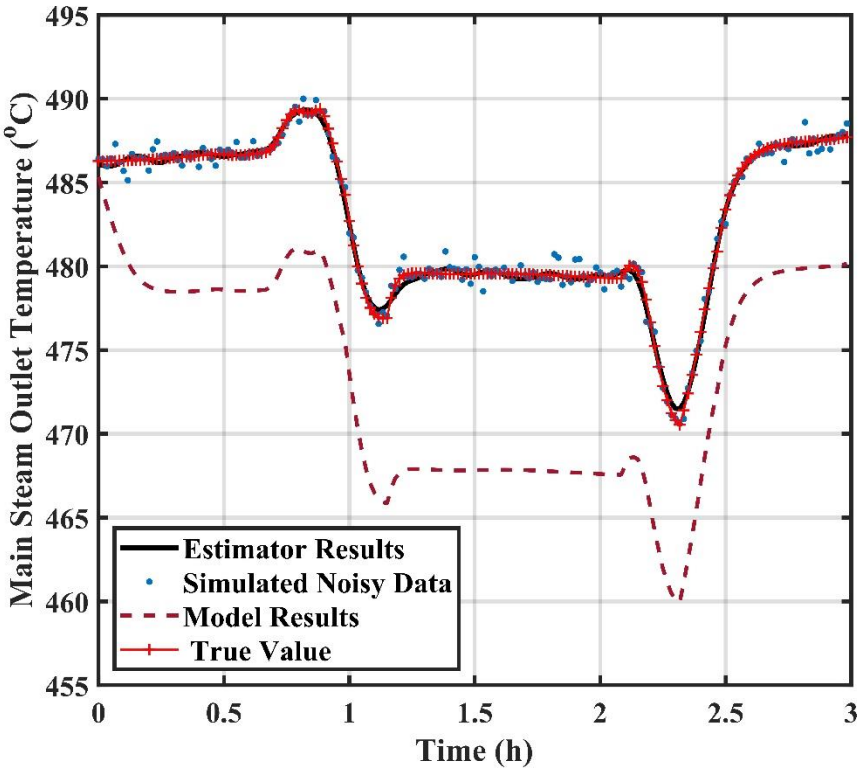


Fig. 2.11. Comparison of the true values of the main steam outlet temperature ($T_{st,out}$, as shown in Fig. 2.3) with the noisy data, model results, and estimator results.

Fig. 2.11 compares the 'true values' of the main steam outlet temperature with the noisy data, model results, and estimator results. It is observed that despite high model mismatch and noise content in the measurement data, estimator results match very well with the true values with the maximum absolute instantaneous error (MAIE) between true and estimator being 1.07 °C. It can be observed that even during the fast transients (i.e., between 0.5-1 h and 2-2.5 h), estimator performance is excellent having an RMSE of around 0.28 °C (between 0.5-1 h) and 0.42 °C (between 2-2.5 h) with an overall value of around 0.25 °C for the entire duration.

Fig. 2.12 shows the results for the outlet flue gas temperature. During the arbitrary random fluctuations shown in Fig. 2.9**,** especially under the impact of the inverse profile in the temperature at 1 h and at 2.5 h, a mismatch between the truth and the estimator results can be observed with a MAIE of 3.4 °C. However, for most of the regime, estimator results have an excellent match with the truth with an overall RMSE value of around 0.56 °C for the entire duration and 0.77 °C (between 0.5-1 h) and 0.91 °C (between 2-2.5 h) during the transients.

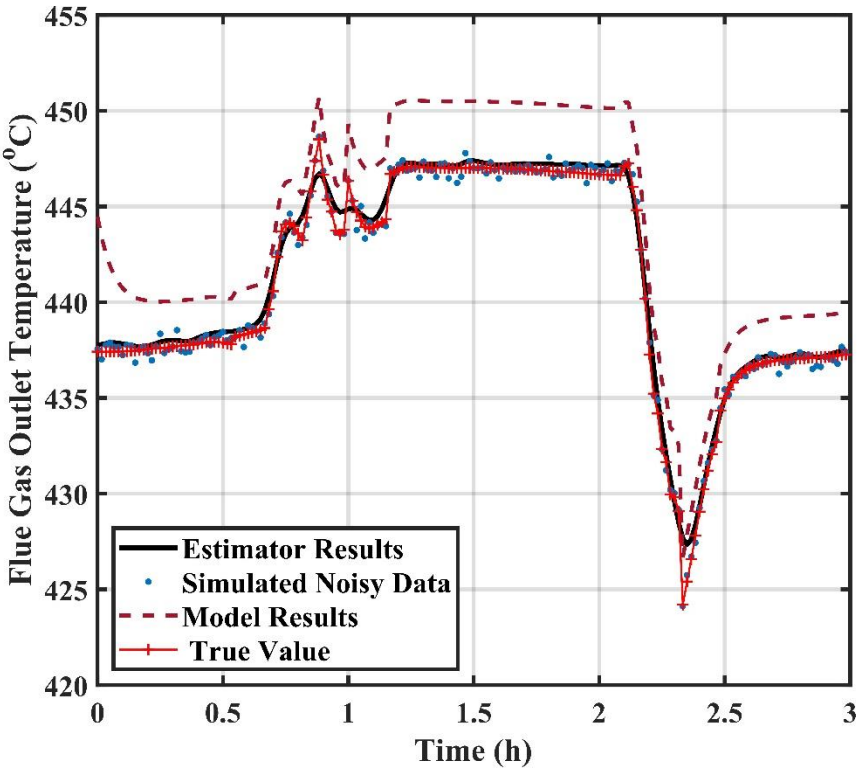


Fig. 2.12. Comparison of the true values of the flue gas outlet temperature ($T_{fg,out}$, as shown in Fig. 2.3) with the noisy data, model results, and estimator results.

The estimator yields results for many unmeasured variables. Fig. 2.13 shows the estimated average steam temperature profiles for each pass of the superheater system along the horizontal direction (see Fig. 2.3). As expected, Pass 6 (the downstream pass) has the lowest steam temperature and Pass 1 (the upstream, front-most pass) has the highest. It is observed that the magnitude of temperature increases between Pass 6 and Pass 5 is the highest due to the temperature difference between the flue gas and steam temperature being the highest there as the relatively cold steam enters at Pass 6. On the other hand, the driving force is the lowest towards the steam outlet side (Pass 1) as the steam temperature increases. Fig. 2.14 shows the estimated average flue gas temperature profiles for each pass of the superheater. It is observed that the temperature dynamics of the flue gas side is much higher than is observed on the steam tube (Fig. 2.13). This is mainly because of the large thermal inertia of the tube metal mass. Estimated profiles of other critical variables, such as the temperature distribution in intermediate headers and return bends for outer tube walls and flue gas temperatures (see Fig. 2.2**),** can be found in **Appendix A.**

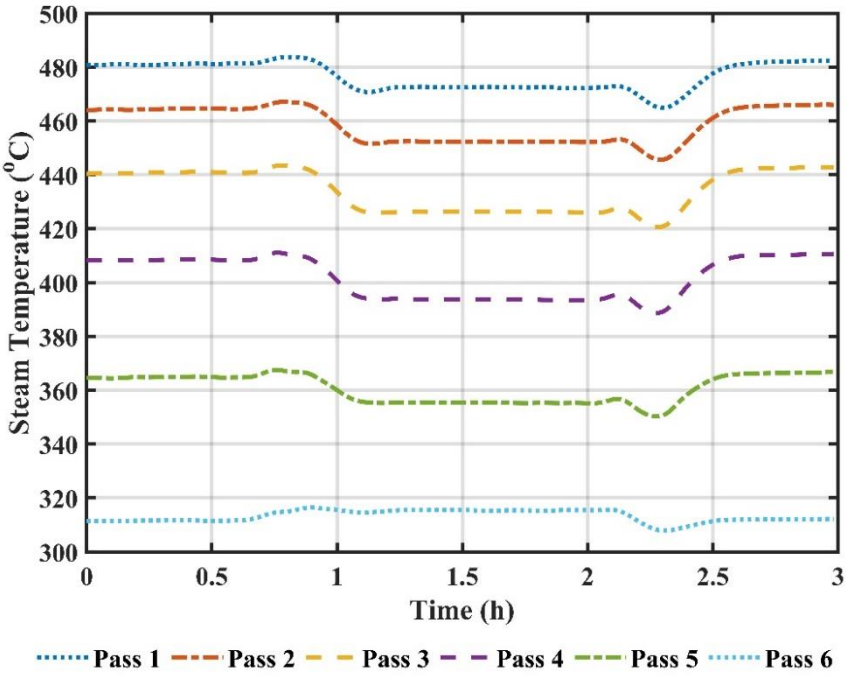


Fig. 2.13. Estimated average steam temperature profile in the horizontal direction (x-direction, as shown in Fig. 2.3) along the superheater.

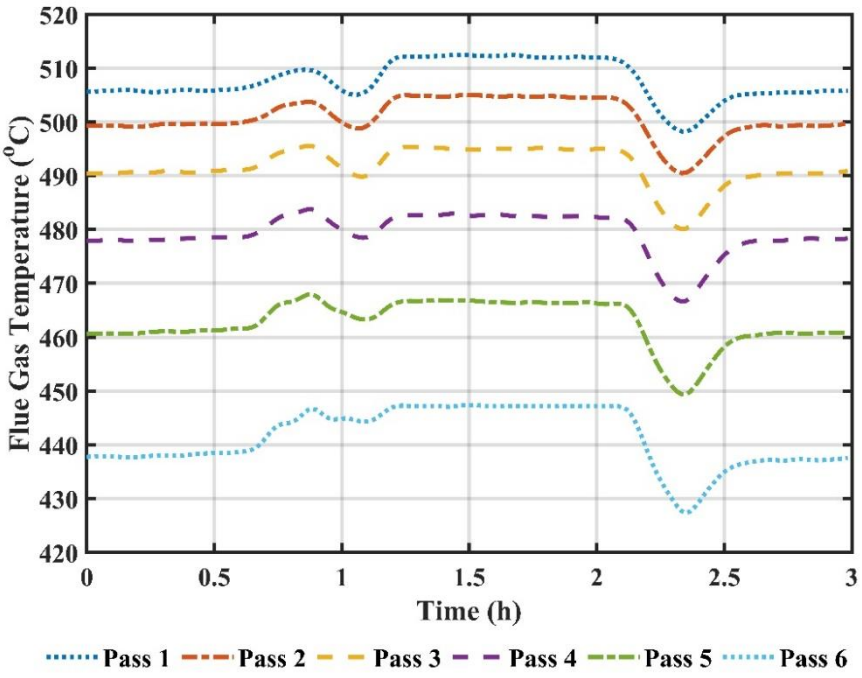


Fig. 2.14. Estimated average flue gas temperature profile in horizontal direction (x-direction, as shown in Fig. 2.3) along the superheater.

One of the critical variables required for the health monitoring of a high-temperature superheater system is the tube temperature profile along the steam flow path due to temperature being a driving factor for elevated temperature damage mechanisms like creep. The temperatures of the tubes at the interior of the superheater are difficult to measure due to the high cost of sensor installation and the difficulty of access for installation and maintenance. However, using a physics-based model allows temperatures at these locations to be calculated. Fig. 2.15 and Fig. 2.16 compare the tube temperatures near the steam outlet (Pass 1) and steam inlet (Pass 6) side. The temperatures along the tube length in the vertical direction are designated by location along the steam flow path based on the control volume discretization scheme shown in Fig. 2.3**.** For example, in Fig. 2.15, the temperatures on the steam outlet side are shown with Location [1,1] nearest to the steam outlet header having the highest tube temperature and Location [6,1] having the lowest tube temperature.

Fig. 2.15 and Fig. 2.16 show that there is not only considerable difference in the tube temperature between passes, but even within the pass. Pass 1 tube temperature profiles are not only much higher than the Pass 6 tubes, but also have steeper temperature transient responses with high overshoots and undershoots. As the thermal stress and therefore tube rupture depends on these temperature profiles [69,70], Pass 1 tubes, especially those near Location [1,1], are desired to be monitored while Pass 6 tubes have considerably much lower risk for failure compared to Pass 1 tubes.

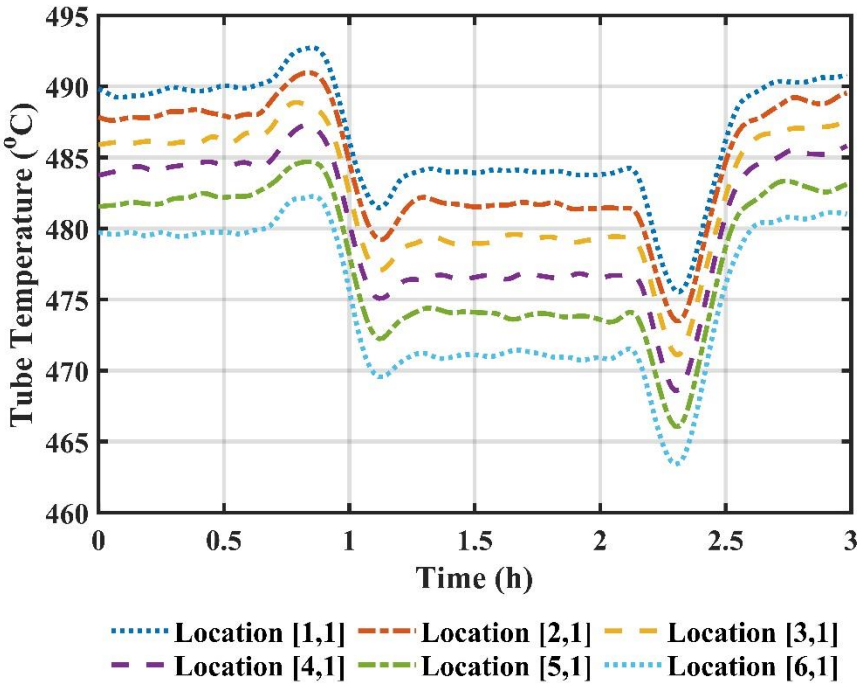


Fig. 2.15. Estimated Pass 1 (near steam outlet) tube temperature profiles in vertical direction (y-direction, as shown in Fig. 2.3).

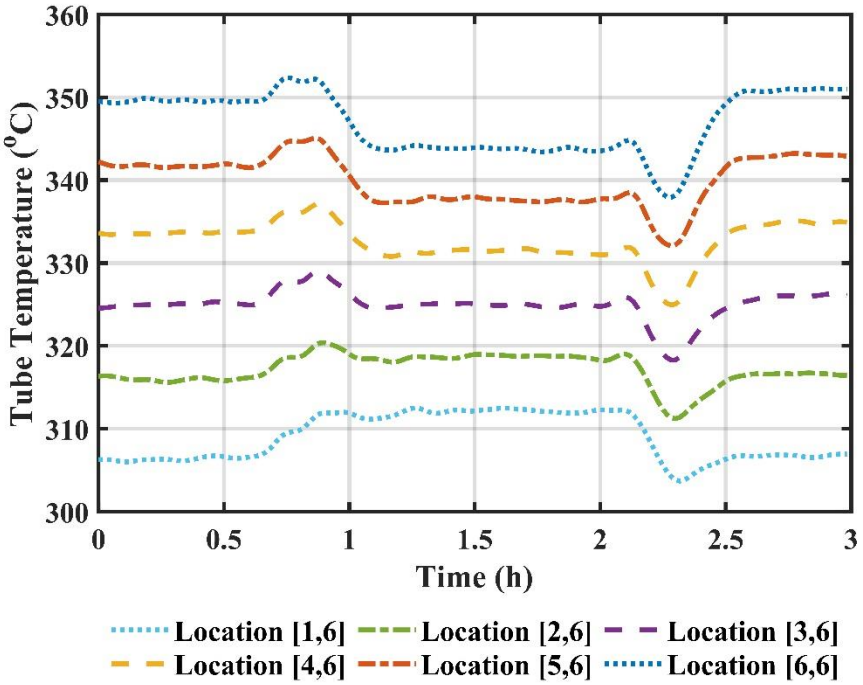


Fig. 2.16. Estimated Pass 6 (near steam inlet) tube temperature profiles in vertical direction (y-direction, as shown in Fig. 2.3).

2.6 Conclusions

This section developed a first-principles 2-D distributed DAE model to simulate the operation of high-temperature boiler components. The developed model is modular so it can be used to configure and simulate various types of geometry and configuration. A modified form of the EKF for the DAE system that can handle uncertainties in differential and algebraic states was used for state estimation. Results from the model and estimator were compared with the operational data for the superheater section of an NGCC power plant. The RMSE for the main steam temperature was found to be 0.2 ^{o}C when the modified estimator was used as opposed to about 1 ^{o}C when the first-principles model was used. Similar improvements in results were also observed when the estimator framework was used for other key state variables essential for health monitoring of the high-temperature components such as tube and flue gas temperatures. Undesired error between the model and measured values was observed for the flue gas and tube temperatures with the instantaneous absolute error being as high as 2.5 ^{o}C observed for the tube temperatures. The estimator was found to reduce the absolute instantaneous error below 0.3 ^{o}C for most variables. Finally, the results were presented for the load-following operation of the plant when the plant load was changed from a lower load to a higher load and then back to the lower load. Performance of the estimator for estimating critical variables like main steam and flue gas temperature was evaluated in the presence of high model mismatch and noisy measurement data. The estimator results matched the true values with a maximum MAIE of 3.4 °C for flue gas and 1 °C for the main steam temperature. The utility of the estimator in predicting overall steam and flue gas temperature profiles and critical tube temperatures was shown, highlighting regions of high excursions in temperature due to load-following operations.

3 Development and Validation of the Bayesian ML Model

3.1 Introduction

Out of many data-driven model building approaches, artificial neural networks (ANNs) have been widely used with outstanding successes in their applications for constructing predictive models even for systems with high level of complexity as well as significant difficulty in model development using first principles[71–73]. Despite these successes, the black-box nature of ANN and similar data-driven models makes it practically impossible to understand the relationship between input and output variables, leading to the lack of model interpretability. In addition, ANN-based models often suffer the curse of dimensionality due to large number of parameters often required for the given architecture [74]. Along with this comes the possibility of model overfitting and the requirement of large amount of data for training such models[75]. Large amount of data may not be available for all processes for varieties of reasons such as harsh (e.g., corrosive, erosive, high temperature) operating conditions, lack of (cheap) technologies for measuring the desired variables, lower sampling frequency of the available measurement technology in the plant or experimental setup, etc. ANN models developed using limited data may have poor predictive capability and high uncertainty[76,77].

3.1.1 Development of Sparse Data-Driven Models Amidst Uncertainties

In the light of the black-box nature of resulting models from many data-driven model development approaches, it is desired that identified models be interpretable and sparse, providing succinct representation of a system. By this we avoid overfitting and modeling spurious characteristics such as noise since overfitting leads to low bias but high variance and therefore undesired for a predictive model. In most cases, developed data-driven approaches are being tested with synthetic data and on the assumption that measurement noise is given by Gaussian distributions that are independent. However, real-life measurement data can contain noise with auto and cross-correlations, and these can result in biased estimates of parameters.

Bayesian inferencing is an approach in system identification that is useful for uncertainty quantification while estimating model parameters. Here, the conditional distribution of the parameter given the data is sought by maximizing posterior probability of the parameters. This approach allows for the incorporation of user belief about the model parameters (e.g., user may have a belief about the activation energy of a chemical reaction) included in the form of prior distribution of the model parameters. While a fully Bayesian approach can be highly computationally expensive, this work develops an approximate Bayesian approach implemented in an Expectation-Maximization framework where model parameters are estimated while explicitly accounting for noise in the data and their possible correlations.
To achieve the goal of interpretability and sparsity, this work proposes a method whereby nonlinearities in system dynamics are approximated by a linear combination of a set of basis functions selected using a developed branch and bound algorithm. The algorithm incorporates a measure of parameter estimability in the model selection criterion greatly reducing computational burden often associated with the branch and bound algorithm.

3.1.2 Mass Conservation by Sparse Data-Driven Models

Predictions from data-driven models often violate conservation laws as these models are at best, approximations of the original system behaviors, and training data often contain noise and/or bias. These factors consequently place significant limitation on the extrapolation capabilities of fully data-driven models. While many physics-informed data-driven approaches have been developed in attempt to address this challenge, the proposed methods are mostly system specific requiring a significant knowledge of details regarding the system in question. However, such information may not be readily available to the model developer. In addition to this, different forms of ensemble models have also been proposed to enhance the capabilities of data-driven models across wider operating ranges. Notwithstanding, frequent retraining of the models may still be unavoidable for systems associated with high level of uncharacterized noise since model trainings are only based on historical data which may lack information about evolving new trends in the system behavior due to uncertainties. Meanwhile, these models cannot guarantee exact satisfaction of conservation rules.

This work proposes two algorithms that guarantee the absolute satisfaction of mass conservation rules by sparse data-driven models. To achieve this goal, the algorithms rely only on known boundary conditions which are generic and readily available. The first algorithm proposed augments the original sparse model with a data reconciliation step while the second algorithm exploits the sparse model structure to formulate a set of equality constraints on the model parameters. The resulting mass-constrained models retain their original sparsity and interpretability while guaranteeing absolute satisfaction of mass conservation laws.

3.2 Identification of Sparse Static and Dynamic Models from Noisy Data

Selection and identification of linear and nonlinear models given some candidate models have been the subject of a number of studies [78–83]. However, an important goal is that the identified model to be sparse, offering a concise representation of the system to prevent overfitting and the inclusion of spurious characteristics like noise. Overfitting results in low bias but high variance, making it undesirable for a predictive model. In addition, candidate basis functions should possess some desired characteristics such as continuity of function and derivatives based on the intended application of the identified model. For steady-state models, these aspects have been addressed by several researchers. Existing literature [84] have noted that the choice of inappropriate model is a major reason for parameter non-identifiability in system identification. Meanwhile several authors [85–88] have pointed out that an unguided increase in the size of basis set of regressors will lead to the model describing noise and other aspects of the training data as the model size becomes undesirably large with several redundant parameters. [89] have developed a comprehensive regression and classification approach for building surrogate steady-state models from a given dataset by solving a mixed integer nonlinear programming problem.

Identification of dynamic models has traditionally been done for application to model-based control approaches [90]. Typical approach in this area is to evaluate several candidate models and select the best model based on some criterion such as root mean squared error (RMSE), Akaike Information Criterion (AIC), etc. Succinct representation of dynamical systems through optimal selection of basis function from several candidate basis functions has been addressed by several authors [91–95]. Diverse candidate models have been considered including finite mixture models

(FMM), radial basis functions (RBF), varieties of autoregressive with exogenous input (ARX) models, Gaussian mixture models (GMM) and linear parameter varying (LPV) models. Algebraic learning via elastic net (ALVEN) has been developed for automatic feature selection in a three-step implementation of the elastic net regression for developing static and dynamic models [96]. ALVEN uses the elastic net regression that uses linear combination of the L_1 and L_2 penalties of the lasso and ridge regression methods. The work by [94] focused on reducing the computational costs associated with high dimensional datasets. [97] have proposed the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm. The SINDy algorithm attempts to discover governing equations for dynamic systems through sparse regression for the selection of minimum features required for the prediction of the dynamic behavior of a given system. It rests on the assumption that only few of a class of basis functions are required for the given system. The SINDy algorithm has been applied to the identification of dynamics of strongly nonlinear systems [92,98] and adaptive model identification [91,99]. The SINDy algorithm has also been used for fault prediction [100,101] and model predictive control [102] including applications to nonlinear two-time scale processes [103]. Meanwhile, for steady-state model building using basis functions, one of the popular approaches is Automatic Learning of Algebraic Models (ALAMO)[87], that employs mixed integer nonlinear programming (MINLP) with adaptive sampling techniques.

One difficulty in identification of dynamic models is presence of noise in the transient data. Presence of noise in the data can lead to high uncertainty in the parameters. Parameter estimates using the SINDy algorithm has been reported to be highly sensitive to noise in the measurement data [94,104]. In addition, as the model structure is unknown and in fact optimal selection of the structure is also an objective of sparse identification, presence of noise in the data can lead to high uncertainty both in the structure and parameters. In general, it is assumed that the measurement data are corrupted with noise that is normally distributed with zero mean and a small variance. When the parameters are estimated using classical parameter estimation methods that seeks to minimize some form of weighted least squares, variances in the measurement data are to be guessed by the user as variances in the measurement data are often unknown or not reported. System identification works well despite expected discrepancy in the user guess if the variances in the measurement data are small, model structure is reasonably known, and there is less uncertainty in parameter estimates. As much of the works discussed before has considered synthetic data (i.e., typically data are generated from a high-fidelity model) and generated the noisy data artificially, sparse identification generally worked well. However, these assumptions are questionable for an industrial process. One alternative approach is Bayesian estimation of parameters that seeks to identify conditional distribution of the parameter given the data by maximizing posterior probability of the parameters. The Bayesian approach facilitates incorporating user belief about the model parameters (e.g., user may have a belief about the activation energy of a chemical reaction) included in the form of prior distribution of the model parameters. However, a fully Bayesian approach that seeks to obtain the optimal posterior distribution given the data is computationally expensive to be employed for real life implementation particularly when the identification may have to be redone with time. An approximate Bayes inference ensues when one seeks to find maximum a posteriori estimate (MAP) of parameters. Expectation maximization (EM) algorithm is a popular approach to solving such problems [105–113]. This approach has been shown to be efficient for parameter estimation using noise-corrupted data [107,114,115] including when noise transients are characterized by a dynamic model [116]. In the past, an EM algorithm was developed for approximate Bayesian estimate of parameters maximizing the joint posterior probability distribution of the state and parameters given

the data [117]. However, similar to other previous works in this area, only one model form was considered. Specifically, an extended bilinear model to represent the nonlinearity was mainly considered. Bilinear and extended bilinear models have been widely investigated in the literature [118–120]. In this recent work [117], the bilinear/extended bilinear models were successfully applied to identification of the Van de Vusse reactor and acid gas removal unit as part of an integrated gasification combined cycle plant. However, bilinear models may be too restrictive to model highly nonlinear processes in general. In addition, similar to other works in this area it was assumed that the measurement noise is given by normal Gaussian distributions that are independent. However, real-life measurement data can be colored, which can cause biased estimates of parameters. Therefore, the purpose of this work is to extend the work in [117] by developing algorithms for sparse identification of dynamic models, and obtain MAP estimate of parameters using measurement data that are corrupted with colored noise.

Consideration of colored noise has been considered in the state estimation literature. Two popular methods are the augmentation approach [121] and the measurement differencing approach [122,123]. In the former approach, the noise is augmented with the actual state of the system and complete state estimation is done employing efficient methods like Kalman filtering. One disadvantage of this method is that it assumes full observability of the system. Also, the approach leads to increased number of states to be estimated resulting in increased dimension of the required filter. In the measurement differencing approach, whitening of the colored noise is attempted by replacing the actual process measurements with a weighted first difference of the present and previous measurements. This approach can eliminate potential ill-conditioning associated with the augmented-state approach [122]. A direct application of the approaches from the state estimation literature is plausible for the current work since states (and/or parameters, if desired) are estimated recursively and the model form is assumed to be given while performing estimation. In this work, we develop a two-stage approach employing the EM algorithm. To begin with, the elements of the noise covariance matrix including the off-diagonal elements are specified as hyperparameters and in the first stage, MAP estimate of parameters given data is sought in a modified measurement differencing approach, while in the second stage, the hyperparameters are updated based on the newly obtained parameters.

Sparse identification requires optimal selection of basis functions, that can lead to a large-scale combinatorial optimization problem. Thus, an efficient method is desired for the underlying integer programming problem. In this work, we use the branch and bound (B&B) algorithm [124] for optimal selection of basis functions. The B&B algorithm has been successfully used for obtaining efficient and globally optimal solutions to subset selection problems [78,125–128] . One critical issue in system identification is the poor estimability of many candidate models [129]. Thus in this work, we include a measure of estimability in the model selection objective through a modified version of the Akaike's Information Criterion (AIC) [130].

3.2.1 Model Approximation

A nonlinear system can be represented by a set of differential and algebraic equations as follows:

$$\dot{x} = f(x, u, \theta) \tag{3.1}$$

$$y = g(x) \tag{3.2}$$

where *x*, *u*, *y* and θ represent the states, inputs, measurements and model parameters respectively. Such models vary in degree of complexity depending on the actual components and processes represented. The general class of nonlinear model considered in this work seeks to approximate the nonlinearity in the system with a linear combination of nonlinear transformations of the input variables and their interactive effects with the measured outputs. Here, the discrete input-output dynamic model of the following form is considered:

$$y_k = \boldsymbol{A} y_{k-1} + \boldsymbol{C} U_{k-1} + \omega_k \tag{3.3}$$

where $y_k \in \boldsymbol{\mathcal{R}^{n\times 1}}$ is the set of measured outputs at time *k*; $\boldsymbol{A} \in \boldsymbol{\mathcal{R}^{n\times n}}$ and $\boldsymbol{C} \in \boldsymbol{\mathcal{R}^{n\times nu}}$ are the model parameters to be estimated, $\omega \in \boldsymbol{\mathcal{R}^{n\times 1}}$ represents the additive noise with covariance matrix C_ε while $U_k \in \boldsymbol{\mathcal{R}^{nu\times 1}}$ denotes the basis functions. In the above, $\boldsymbol{n}$ represents the number of measured outputs while $\boldsymbol{nu}$ is the number of basis functions obtained from the transformations of the input variables. In addition to constants and linear terms (e.g., u_i), several nonlinear basis functions are considered as candidates such as inverse (e.g., $\frac{1}{u_i}, \frac{1}{u_i^2}, \frac{1}{\sqrt{u_i}}$), square root (e.g., $\sqrt{u_i}$), polynomial and multinomial (e.g., $u_i^2, u_i y_j, u_j u_l$), sigmoidal (e.g., $\frac{1}{1+e^{-u_i}}$), exponential and logarithmic (e.g., $e^{-u_i}, log(u_i)$) functions. These candidate basis functions lead to explainable candidate models in which the outputs can be understood as interpretable transformations of the input variables. For example, the constant term can represent any fixed value of the output that does not depend on the inputs, the linear terms represent the direct linear effects of the input variables such as the temperature difference and concentration difference driving force, the exponential term can represent Arrhenius-type relationship for reaction kinetics, and so on.

3.2.2 Model Estimation

Bayesian inferencing

The Bayes' rule represents the posterior distribution of the model parameters as a product of the likelihood of the data and the prior distribution of the parameters:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{3.4}$$

Here, $p(y)$ is the normalizing function given by the following equation:

$$p(y) = \int_{\Theta} p(y|\theta)p(\theta)d\theta \tag{3.5}$$

The goal for system identification is to obtain optimal model parameter values and estimate the noise covariance from a given noisy data. Here we employ the expectation-maximization (EM) algorithm in which the noise covariance components are treated as the hyperparameters that are

estimated simultaneously with the model parameters. The EM algorithm proposed by [131,132] has been noted as a generic procedure for iterative joint estimate of parameters and hyperparameters [133]. Full details on how this algorithm has been applied for input-state-output model estimation from data corrupted with gaussian noise are available in [117]. In this work, the capability of the algorithm has been extended from estimation of the extended bilinear model to a general case of discrete input-output nonlinear model with a large number of candidate basis functions. The algorithm has also been extended to estimate parameters when the measurement data are corrupted with colored noise. A brief highlight of the current algorithm is given below pointing out the modifications from what is presented in in the previous work in [117].
Let the model prediction at time k be depicted as $h(u_{k-1}, y_{k-1}, \theta)$ and the true underlying signal, $\hat{y}_k$. Then the expressions for measured outputs are given in the general form

$$y_k = h(u_{k-1}, y_{k-1}, \theta) + \omega_k \tag{3.6}$$

Using Bayes' law, the joint probability distribution of the raw outputs (without noise) and model parameters given the measurements is given by:

$$p(\hat{y}, \theta | y) = \frac{p(y|\hat{y}, \theta) p(\hat{y}|\theta) p(\theta)}{p(y)} \tag{3.7}$$

We seek to find:

$$\hat{y}^*, \theta^* = \underset{\hat{y},\theta}{argmax}\, p(\hat{y}, \theta | y) \tag{3.8}$$

Noise characterization

Overall, it is assumed that noises in individual measured variables are correlated with one another and are not necessarily identically distributed. One approach is to parameterize the unknown measurement noise covariance with hyperparameters λ_q as follows:

$$C_\varepsilon = \sum_q \lambda_q V_q \tag{3.9}$$

The hyperparameters, λ_q scale the contribution of the basis matrices V_q to the noise covariance. In the previous work [117], the matrix V_q is sparse with the q^{th} element in the leading diagonal equal to one and all other elements set to zero. For the case of correlated noise, off-diagonal elements in the sparse matrices V_q are not set to zero so that cross-correlation, if present, between noise in different measurements can be estimated.

Expectation-Maximization (EM) algorithm

The first step in the EM algorithm is the E-step. Denoting the conditional expectation of the parameters at the i^{th} iteration as $\eta^{(i)}_{\theta|\hat{y}}$, it was shown in [117] that a gradient ascent on the joint log likelihood function $ln(p(\hat{y}, \theta|y))$ can be used for a recursive identification and estimation of the model parameters as follows:

$$\eta_{\theta|\hat{y}}^{(i+1)} = \eta_{\theta|\hat{y}}^{(i)} - \left(\frac{\partial^2 ln\big(p(\hat{y},\theta|y)\big)}{\partial\theta^2}\right)^{-1} \frac{\partial ln\big(p(\hat{y},\theta|y)\big)}{\partial\theta} \tag{3.10}$$

Using Taylor series approximation at the current estimate of θ:

$$\hat{y} - h\big(u,y,\eta_{\theta|\hat{y}}\big) \approx \frac{\partial h\big(u,y,\eta_{\theta|\hat{y}}\big)}{\partial\theta}\Delta\theta + \omega \tag{3.11}$$

Let

$$J = \frac{\partial h\big(u,y,\eta_{\theta|\hat{y}}\big)}{\partial\theta} \tag{3.12}$$

$$\bar{y} = \begin{bmatrix} \hat{y} - h(u,y,\eta_{\theta|y}^{(i)}) \\ \eta_\theta - \eta_{\theta|y}^{(i)} \end{bmatrix} \tag{3.13}$$

$$\bar{J} = \begin{bmatrix} J \\ I \end{bmatrix} \tag{3.14}$$

$$\bar{C}_\varepsilon = \begin{bmatrix} \sum \lambda_q V_q & 0 \\ 0 & C_\theta \end{bmatrix} \tag{3.15}$$

Then

$$\eta_{\theta|\hat{y}}^{(i+1)} = \eta_{\theta|\hat{y}}^{(i)} + \Delta\eta_{\theta|\hat{y}} \tag{3.16}$$

where

$$\Delta\eta_{\theta|\hat{y}} = C_{\theta|\hat{y}}(\bar{J}^T\bar{C}_\varepsilon^{-1}\bar{y}) \tag{3.17}$$

$$C_{\theta|\hat{y}} = (\bar{J}^T\bar{C}_\varepsilon^{-1}\bar{J})^{-1} \tag{3.18}$$

Now in the M-step, the estimation of the unknown noise covariance represented by hyperparameters in the E-step is done by maximizing the likelihood of the conditional estimate of the data obtained from the E-step on the current estimate of the hyperparameters. More details of the derivations of the following resulting equations can be found in [117] and [134].

$$ln\big(p(\hat{y}|\lambda)\big) = ln \int q(\theta)\frac{p(\hat{y},\theta|\lambda)}{q(\theta)}d\theta \tag{3.19}$$

By introducing a tractable function $\bar{F}$, using Jenson's inequality,

$$ln \int q(\theta) \frac{p(\hat{y}, \theta|\lambda)}{q(\theta)} d\theta \geq \bar{F} = \int q(\theta)\, ln \frac{p(\hat{y}, \theta|\lambda)}{q(\theta)} d\theta \tag{3.20}$$

and a gradient ascent on the log function $\bar{F}$ ensues:

$$\eta_{\lambda|\hat{y}}^{(i+1)} = \eta_{\lambda|\hat{y}}^{(i)} - \alpha \left(\frac{\partial^2 \bar{F}}{\partial \lambda^2}\right)^{-1} \frac{\partial \bar{F}}{\partial \lambda} \tag{3.21}$$

$$M = \bar{C}_{\varepsilon}^{-1} - \bar{C}_{\varepsilon}^{-1} \bar{J} C_{\theta|y} \bar{J}^T \bar{C}_{\varepsilon}^{-1} \tag{3.22}$$

$$\left(\frac{\partial \bar{F}}{\partial \lambda_j}\right) = -\frac{1}{2} tr\{MV_j\} + \frac{1}{2} \bar{y}^T M^T V_j M \bar{y} \tag{3.23}$$

$$\left(\frac{\partial^2 F}{\partial \lambda_{jk}^2}\right) = -\frac{1}{2} tr\{MV_j MV_k\} \tag{3.24}$$

An outline of the above algorithm is given in Algorithm 1.

Algorithm 1 Expectation Maximization for model parameter estimation

For a given training input-output dataset $\{u, y\}$ and model structure with parameters θ

Initialize $i = 0$ and $\eta_{\lambda|\hat{y}}^{(0)}$, and assume priors, η_{θ} on θ

while $\left\|\Delta\eta_{\theta|\hat{y}}\right\| > \varepsilon_1$

E-step: *Gradient ascent on* $ln\big(p(\hat{y}, \theta|y)\big)$

Compute $\Delta\eta_{\theta|\hat{y}}$ based on current estimate of $\bar{C}_{\varepsilon}$ given by $\eta_{\lambda|\hat{y}}^{(i)}$

Update model parameters $\eta_{\theta|\hat{y}}^{(i+1)} \leftarrow \eta_{\theta|\hat{y}}^{(i)} + \Delta\eta_{\theta|\hat{y}}$

M-step: *Maximization of* $ln\big(p(\hat{y}|\lambda)\big)$

Compute $\Delta\eta_{\lambda|\hat{y}}$ based on tractable form of $ln\big(p(\hat{y}|\lambda)\big)$ in Eq.(3.19)- (3.20)

Update $\eta_{\lambda|\hat{y}}^{(i+1)}$ according to Eq.(3.21)

end while

3.2.3 Dynamic Model Selection

It is desired to select the smallest subset of basis functions that best describes the underlying behavior of the system given noisy input/output data. In this section, the criterion for selection of sparse candidate model and the algorithm for selection are discussed.

Model selection criterion

Model selection criterion is an important decision for obtaining the sparse model given the noisy data. Thus, minimization of empirical error by using measures such as mean squared error (MSE) or root mean square error (RMSE) is not sufficient for sparse model development since overfitting should be penalized. Here we employ information criterion-based approach for penalizing overfitting. One of the popular measures in this regard is the Akaike Information Criteria (AIC) [130], which is a measure of the Kullback-Leibler divergence between an underlying true distribution and the estimate of the distribution obtained from a candidate model. Another popular approach is the Bayesian Information Criterion (BIC) which is a measure of the posterior model probability. For limited data, an additional term for small sample bias correction is added to AIC to yield AICc. This is based on the underlying assumption of the existence of a standard regression model for which sampling distribution of the estimated parameters is normally distributed around the true parameter values [129].

$$AIC_c = -2\ln\left(L\left(\hat{\theta}\right)\right) + 2r + \frac{2r(r+1)}{p-r-1} \tag{3.25}$$

In Eq. (3.25), $L\left(\hat{\theta}\right)$ is the likelihood function of the pdf $p(\hat{y})$ evaluated at maximum likelihood parameter values $\hat{\theta}$; r is the number of parameters in model and p is the sample size. Another measure for rewarding sparse models is Information Complexity Criterion (ICC) that seeks to quantify model complexity through functions of covariance of parameter estimates. The trace and determinant of the covariance matrix $\boldsymbol{\Sigma}$ of the estimated parameters can be employed to measure parameter redundancy and estimation instability. Detailed discussion on the extension of popular information criteria such as the AIC and BIC can be found in [129]. In this work, we use the following goodness-of-fit measure:

$$\text{AICc} + 2\text{Cev}(\Sigma) = \text{AICc} + \sum_{i=1}^{r} \ln(\sigma_i^2) - \ln(|\Sigma|) \tag{3.26}$$

where $\boldsymbol{\Sigma}$ is the model parameter covariance and σ_i^2 is the variance of i^{th} parameter.

Branch and bound algorithm

The branch and bound (B&B) algorithm is a popular algorithm for integer programming [124]. Here, this algorithm is used to select the sparse model with the maximum goodness-of-fit measure where model parameters are estimated using the EM algorithm described in Section 3.2.2. There are several reasons that motivate the use of B&B algorithm for the sparse model identification problem considered in this work. The B&B algorithm can reach the global optimum for the discrete decision variables (i.e., selection of globally optimal basis functions), helps to exploit specific characteristics of the objective functions and/or constraints thus pruning large number of candidate

nodes (as will be discussed below in more details), evaluation of specific basis function based on a ranking strategy to rapidly yield good incumbent, and to yield not only one model as the optimal but a number of models that are ranked. To elaborate on the last point, a user can apply some posteriori selection criteria or even select between two combinations of basis functions with very close values of objective functions.

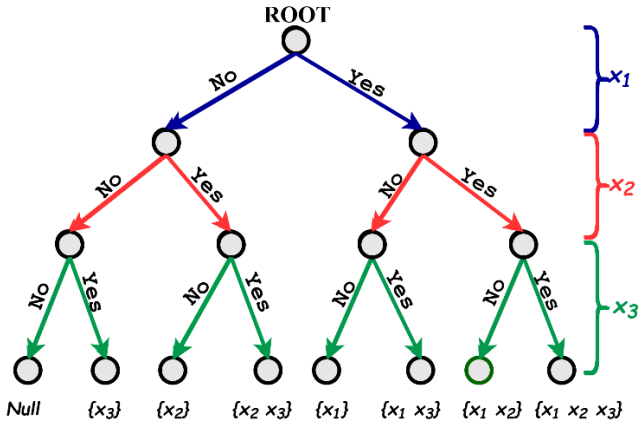


Fig. 3.1. Decision tree for model selection

Fig. 3.1 shows the typical decision tree for the B&B algorithm, the root represents the empty set which expands by element-by-element addition of basis functions moving from one node to another down each branch. Progressing from each node, the decision is made either to add a particular basis function to a parent set or to skip it. In this approach, any expansion in a node on the right represents the addition of a basis function to the selected set while moving along the left branch skips the addition of the current basis function thereby keeping the parent set unchanged. The bottom nodes of the tree are terminal nodes which represent all possible subsets of the universal set. An exhaustive search of this form often results in an intractable problem as the number of variables increases. This is avoided in this work by implementing pruning strategies discussed below.

- **A priori analyses**

This stage involves the ranking of the basis functions in the library. The optimal value of the objective function for individual basis functions is obtained using the EM algorithm. The basis functions are then ranked based on their relative values of objective function. At the top level of the B&B algorithms and for nodes at subsequent levels, the ranking is utilized by evaluating the top-ranked basis function at that level/node, i.e., by considering the first basis function in the set generated by subtracting the set of basis functions that has already been selected from the set of ranked basis functions. This ensues to the best-first approach.

- **Pruning strategies**

Beginning from top rank basis, combinations of basis functions are considered in a systematic way progressing down each branch of the decision tree. In a classical subset selection problem where $X_p = \{x_i | \ i = 1, 2, \dots, p\}$ is the universal set of p elements from which a subset X_q with q elements is to be selected, there are $C_q^p = \frac{q!}{(p-q)!p!}$ candidate subsets and this number increases dramatically as p and q increase, hence the need for an effective pruning strategy in the search for the globally optimal subset without an exhaustive search.

For the case of correlated noise in Eq.(3.9), the model in Eq. (3.3) is linear in all parameters, hence, the Jacobian matrix is independent of the parameter values. This special structure is exploited in determining the estimability of the parameters of a candidate model given the available data by performing a priori rank analyses on the Jacobian matrix which is simply constructed from the input/output data and their transformations. A rank-deficient Jacobian indicates that a candidate model cannot be estimated from the given data, hence the branch is pruned. Following this, a monotonic decrease in the objective function is accomplished as shown in Fig. 3.2.

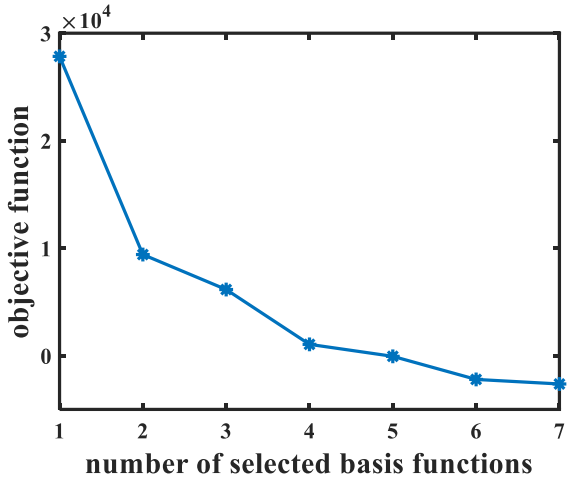


Fig. 3.2. Monotonic decrease in cost function down a branch

The above a priori analyses and pruning process effectively reduce the number of candidate models considered by systematically allowing for the exploration of only promising branches. Further still, the resulting monotonic decrease in the value of the cost function on addition of new basis to the selected set as shown in Fig. 3.2 is exploited for obtaining the sparse model. The termination criterion is either when all candidate nodes are exhausted or when the relative improvement in the objective function is below the specified tolerance. Finally, the algorithm is developed such that a desired set of hierarchically ranked candidate models is returned leaving the user with the option of choosing any top rank model that yields succinct representation of the system with explainable basis functions.

An outline of the algorithm for a specific branch of the decision tree is given in Algorithm 2. By incorporating the Bayesian Machine Learning (BML) approach in Algorithm 1 for parameter estimation and the Branch and Bound approach in Algorithm 2 for model selection, the entire Bayesian Identification of Dynamic Sparse Algebraic Model (BIDSAM) algorithm is obtained for building the desired number of hierarchically ranked sparse models. The overall algorithm is

shown in the flowchart given in Fig. 3.3.

Algorithm 2 Branch and Bound for enumeration and selection of appropriate model structure

Given a family of (hierarchically ranked) candidate basis functions $\boldsymbol{U_{ranked}} \in \boldsymbol{\mathcal{R}}^{nu \times 1}$
Begin with $\emptyset$ basis functions
Start at $l = 1$
while $l < nu$ or $\Delta(\boldsymbol{AICc} + \boldsymbol{2Cev}(\boldsymbol{\Sigma})) > \varepsilon_2$
  Add $\boldsymbol{U_{ranked}}(l)$ basis function to candidate model
  If $\boldsymbol{J}$ in Eq. (3.12) is rank deficient
    break
  else
    obtain θ^* and $\boldsymbol{AICc} + \boldsymbol{2Cev}(\boldsymbol{\Sigma})$ by **Algorithm 1**
    Compute $\Delta(\boldsymbol{AICc} + \boldsymbol{2Cev}(\boldsymbol{\Sigma}))$
  end if
  $l = l + 1$
end while
Add model to library of candidate models

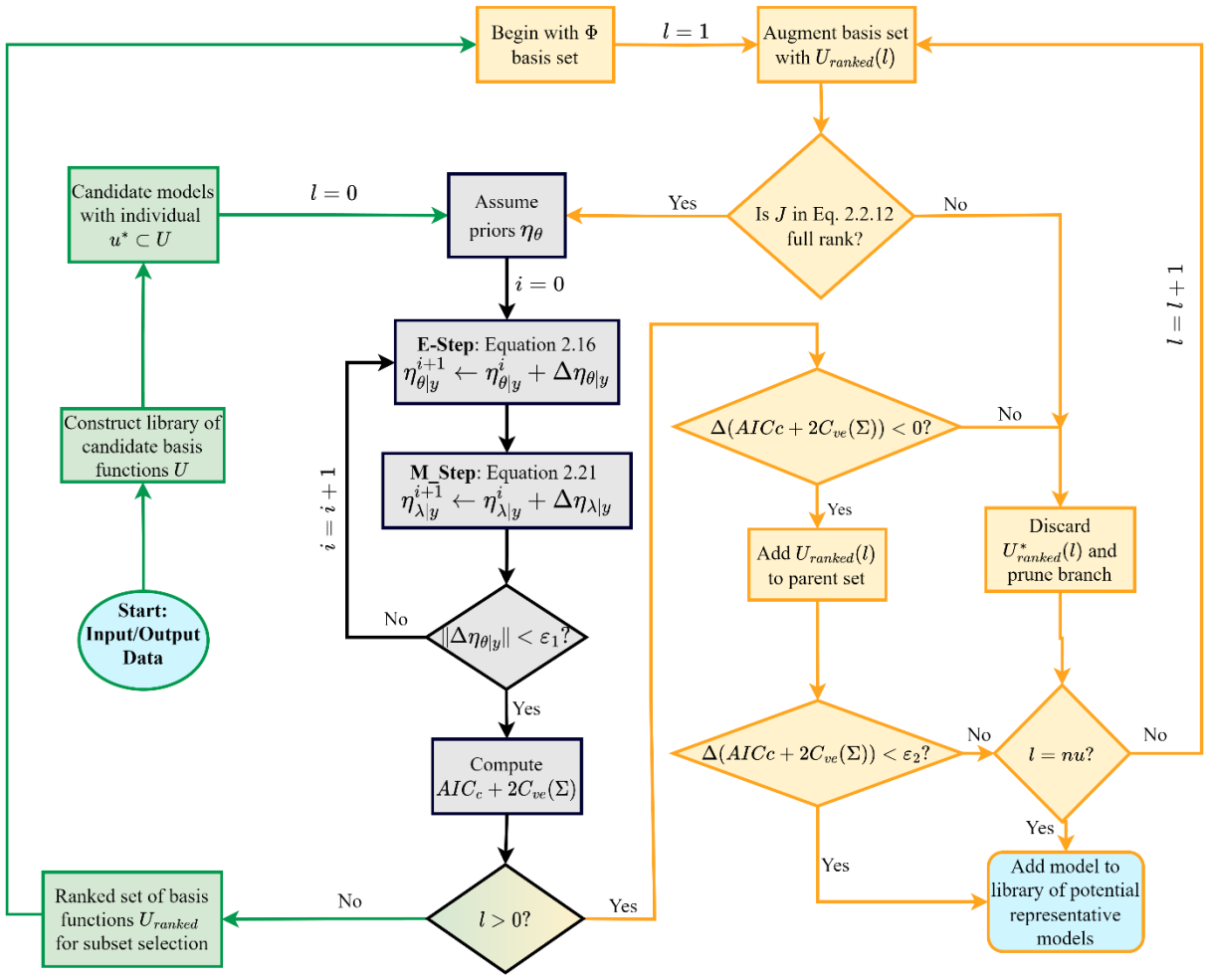


Fig. 3.3. Detailed flowchart for the Bayesian Identification of Dynamic Sparse Algebraic Model (BIDSAM) algorithm.

3.3 Applications of the BIDSAM algorithm

This section presents results from the evaluation of the developed BIDSAM algorithm under varying scenarios. The algorithm is tested by using it for both steady state and dynamic model identification from synthetic and real plant data. Models are developed for applications both as fully data-driven model as well as in hybridization with first-principles models. Overall, two other criteria used for comparing the performance of the models at different points are the AIC_c described in Eq.(3.27) and the $\%RMSE$ given below:

$$\%RMSE = \sqrt{\frac{1}{k_n} \times \sum_{k=1}^{k_n} \left(\frac{y_k - h(u_{k-1}, y_{k-1}, \theta)}{y_k} \times 100 \right)^2} \tag{3.27}$$

where k_n is the total number of time steps considered. In the cases where synthetic data were used, y_k is the simulated data from the differential equations modeling the respective systems at the time instant k, otherwise, y_k denotes the measured plant data.

3.3.1 Steady state modeling

This section presents results in which BIDSAM algorithm is used for development of steady state models. Synthetic data are used in this case and models obtained from the BIDSAM algorithm proposed in this work are compared with those from the Automatic Learning of Algebraic Models (ALAMO)[87] software. For the BIDSAM algorithm, basis functions as listed above are used. For ALAMO, the inbuilt basis functions are used for what is referred to as *ALAMO A*, while for *ALAMO B*, custom basis functions as in the case of the BIDSAM algorithm are used. It should be noted that the tolerance value for convergence in all cases was set to 1*e*-6. In ALAMO, there are several possible performance measures that can be selected. For the results presented in this work, $AICc$ was selected as the objective function for ALAMO.

The case study considered is the nonisothermal Van de Vusse reactor [135] which produces cyclopentanol (B) from cyclopentadiene (A) by acid-catalyzed electrophilic addition of water in dilute solution. Other products resulting from this reaction are cyclopentanediol (C) and dicyclopentadiene (D).

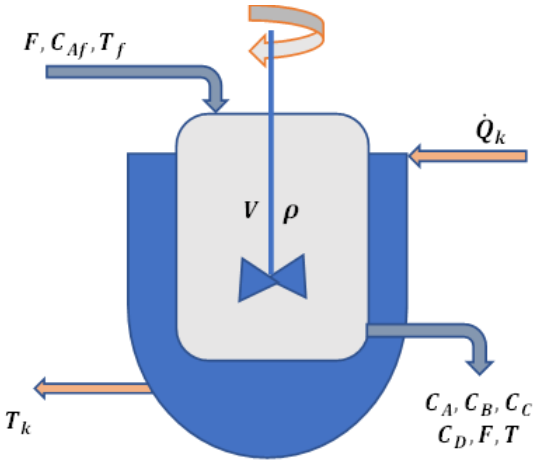


Fig. 3.4. Schematic diagram of non-isothermal van de Vusse reactor

The system consists of a continuously stirred tank reactor with an outer cooling jacket removing the heat of reaction. The inlet to the reactor is a stream of component A at a concentration of C_{Af} and temperature T_f. The chemical reactions proceed as follows:

$$A \overset{k_1}{\rightarrow} B \overset{\kappa_2}{\rightarrow} C \tag{3.28}$$

$$2A \overset{\kappa_3}{\rightarrow} D \tag{3.29}$$

Component material balance equations are given below:

$$\frac{dC_A}{dt} = \frac{F}{V}\left(C_{Af} - C_A\right) - \kappa_1 C_A - \kappa_3 {C_A}^2 \tag{3.30}$$

$$\frac{dC_B}{dt} = -\frac{F}{V}C_B + \kappa_1 C_A - \kappa_2 C_B \tag{3.31}$$

$$\frac{dC_C}{dt} = -\frac{F}{V}C_C + \kappa_2 C_B \tag{3.32}$$

$$\frac{dC_D}{dt} = -\frac{F}{V}C_D + \frac{1}{2}\kappa_3 {C_A}^2 \tag{3.33}$$

The overall energy balance for the reactor and the cooling jacket are given by:

$$\frac{dT}{dt} = \frac{F}{V}\left(T_f - T\right) + \frac{\kappa_w A_r}{\rho C_p V}\left(T_k - T\right) - \frac{\kappa_1 C_A \Delta H_R^{AB} + \kappa_2 C_B \Delta H_R^{BC} + \kappa_3 C_A^2 \Delta H_R^{AD}}{\rho C_p} \tag{3.34}$$

$$\frac{dT_k}{dt} = \frac{1}{m_k C_{pk}}\left[\dot{Q}_k + \kappa_w A_r (T - T_k)\right. \tag{3.35}$$

In the above equations, C_i stands for the concentration of species i, T represents the reactor temperature, T_k denotes the coolant temperature and $\kappa_i's$ are the rate constants given by the Arrhenius equation:

$$\kappa_i(T) = \kappa_{0i}\, exp\left(\frac{-E_{Ai}}{R(T + 273.15)}\right) \tag{3.36}$$

The desired product is B, hence the concentrations of interest are C_A and C_B. Also measured are the reactor temperature, T and coolant temperature, T_k. The manipulated variables are the feed flowrate F and the amount of heat removed by the coolant $\dot{Q}_k$. One source of disturbance to this reaction system is variation in the feed concentration, C_{Af}. For the data-driven model development, three inputs ($\frac{F}{V}$, $\dot{Q}_k$ and C_{Af}) and four measured outputs (C_A, C_B, T and T_k) are considered. Values for the model parameters and operating conditions shown in Table 3.1 are obtained from [136].

The data used for model building is obtained by perturbing the system randomly within the range of $\pm 20\%$ deviation of the steady state values of the manipulated/disturbance variables $\left(\frac{F}{V}, \dot{Q}_k, C_{Af}\right)$ from their steady state values. Noisy data used for model estimation is obtained by the addition of correlated noise with different variance in the different variables to the 'true' data. Model building was done by using randomly selected 60% of the data ((120 out of 200 data points) and using the same data for model building for both the BIDSAM algorithm and the ALAMO software.

Table 3.2 shows the performance measure of the obtained models while Fig. 3.5 show a few validation plots for each model trained from the true data and from the noisy data. The training performance indices, selected basis functions for each model and more validation plots are given in the supplementary material of the published article [137].

Table 3.1. Van de Vusse reactor parameters

Variable	Definition
F	Feed flow rate
V	Reactor volume
C_{Af}	Concentration of species A in feed
T_f	Feed temperature
C_p	Specific heat capacity of reacting mixture
$\kappa_{0_{1,2}}$	Rate constant for reactions 1 and 2
κ_{0_3}	Rate constant for reactions 3
$\dot{Q}_k$	Amount of heat removed by coolant
m_k	Mass of coolant
C_{pk}	Specific heat capacity of coolant
A_r	Reactor surface area
ΔH_R^{AB}	Heat of reaction for conversion of species A to B
ΔH_R^{BC}	Heat of reaction for conversion of species B to C
ΔH_R^{AD}	Heat of reaction for conversion of species A to D

In Table 3.2, RMSE values for all validation cases (i.e., irrespective of whether true or noisy data are used for training) are reported with respect to the true data. In all cases, $\%RMSE$ values for all data (training and validation) are kept below 1%. The maximum $\%RMSE$ value for training and validation data for BIDSAM model is 0.81 and 0.85% respectively while for ALAMO, we have 0.72 and 0.73% respectively. While the ALAMO models yielded relatively lower $\%RMSE$ than the BIDSAM model, the BIDSAM models yield approximately the same level of accuracy as the ALAMO models with number of parameters less than 50% (i.e., more parsimonious model) compared to the ALAMO models. Therefore, the BIDSAM models are found to yield better AIC_c values both for the training and validation datasets. It is also observed in Table 3.2 that the RMSE values obtained by using the BIDSAM models do not change much whether they are trained by using true or noisy data sets, but a significant increase in RMSE values is observed for the ALAMO model when they are trained by using the noisy data compared to the true data.

Table 3.2. Performance measures of BIDSAM and ALAMO models (trained using true data and noisy data) with respect to true data

Model	Model Building Time	Number of Parameters	%RMSE				AIC_c
			C_A	C_B	T	T_k	
True Data							
BIDSAM	18 min	16	0.68	0.71	0.09	0.10	-262.4
ALAMO A*	2.6 sec	43	0.06	0.05	0.01	0.01	-121.0
ALAMO B*	1.13 min	39	0.06	0.05	0.01	0.01	-156.1
Noisy Data							
BIDSAM	17 min	16	0.85	0.82	0.12	0.11	-258.2
ALAMO A*	2.6 sec	23	0.67	0.73	0.13	0.20	-234.8
ALAMO B*	1.4 min	25	0.61	0.73	0.14	0.22	-226.3

A* – Estimated with inbuilt basis functions B* – Estimated with custom basis functions

However, as the number of parameters gets reduced for ALAMO models when they are trained by using the noisy data, their AIC_c values gets considerably improved compared to when they are trained by using the true data. Considering the validation plots given Fig. 3.5 (a-c) for the models trained by using the noisy data, it is observed that the ALAMO models underpredict C_B for higher values of C_B (i.e., higher conversion of C_A), but no such trend could be observed for the BIDSAM model.

In Fig. 3.6 (a-c), model results for cooling water temperature obtained by using each algorithm when model building is done by using the true data as well as the noisy data are compared. As expected based on the RMSE values given in Table 3.2, Fig. 3.6 a shows that the model results are reasonably similar for the BIDSAM model irrespective of whether they are built by using the true data or noisy data, but the differences in model results become obvious in Fig. 3.5 b and Fig. 3.5c for both ALAMO models when they are trained by using the true data or the noisy data. **Table B. 1** in **Appendix B** shows the set of basis functions for each model where the sparsity of the BIDSAM model is obvious.

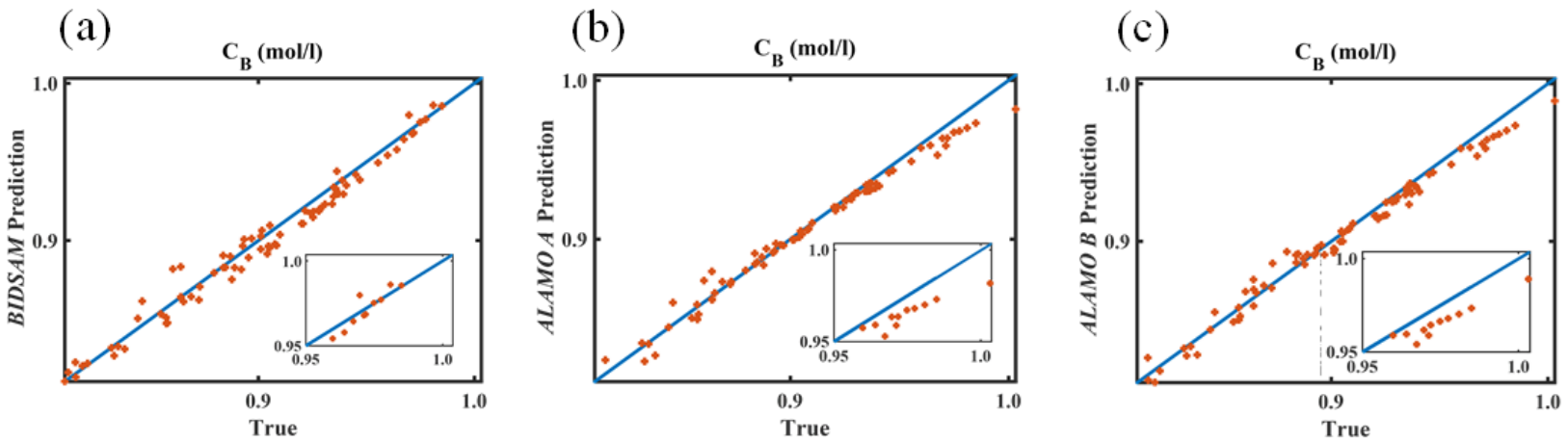


Fig. 3.5. Validation plots for C_B (a) BIDSAM model (b) ALAMO A model (c) ALAMO B model.

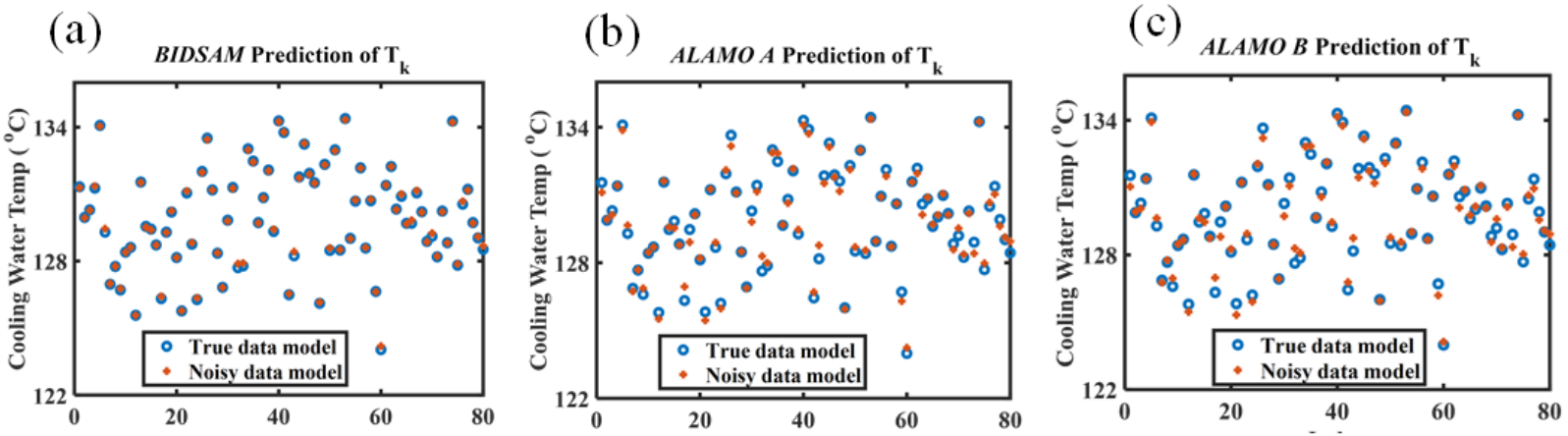


Fig. 3.6. Comparison of models identified from true and noisy data (a) BIDSAM (b) ALAMO A (c) ALAMO B.

Plots of the magnitudes of the coefficients of the BIDSAM model and sparsest ALAMO model while trained using the noisy data are shown in Fig. 3.7. It is interesting to observe that there is considerably large variability in the coefficients of the ALAMO model compared to the BIDSAM model. The absolute minimum and maximum values of coefficients in the BIDSAM model are 0.0772 and 1.823, respectively while for the ALAMO model, those values are 0.0103 and 151.8208, respectively.

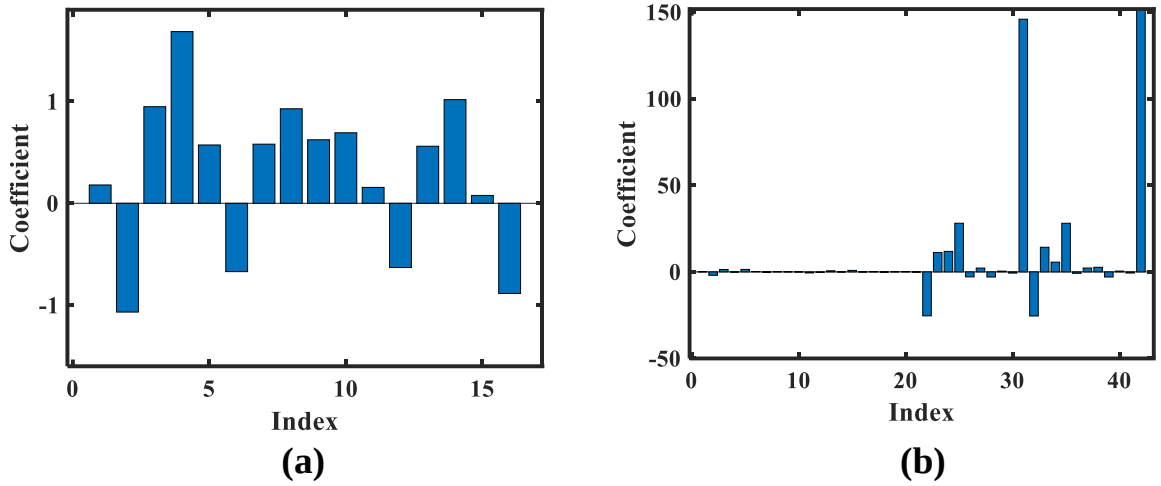


Fig. 3.7. Plot of magnitudes of model coefficients (a) BIDSAM (b) ALAMO.

Considering the training time reported in **Table 3.2**, although the BIDSAM algorithm took the longest time, a breakdown of this time for the model trained with true data as shown in **Table B. 1** in **Appendix B** shows that the most time-consuming step is the search for optimal model from the large number of all candidate models. However, it is important to note that the efficient pruning strategies employed in this work has led to the exploration of only 0.0062% of all possible nodes in this problem.

3.3.2 Dynamic Model Building

This section is aimed at evaluating the performance of the BIDSAM algorithm for real process data in which the characteristics of the noise present in the training data are unknown unlike in the previous cases considered. Data from a CO_2 absorber of a post-combustion CO_2 capture pilot plant are considered.

Fig. 3.8 shows a schematic diagram of the absorber unit decoupled from the process flow diagram of the National Carbon Capture Center (NCCC) pilot plant in Wilsonvile, Alabama, USA has been presented in Figure 2 of [138]. Rigorous thermodynamic modelling, uncertainty quantification, process modeling, and dynamic data reconciliation for the pilot plant data have been reported in previous works [138–140]. This process exhibits high nonlinearity. Following variables are considered:

- ***Input variables (u)***: Flue gas flow rate (lb/hr), Flue gas CO_2 concentration (% dry basis), Flue gas temperature (^oF), Lean solvent inlet temperature (^oF), Lean solvent inlet flowrate (lb/hr).
- ***Output variables (y)***: Flue gas outlet temperature, Flue gas CO_2 concentration.

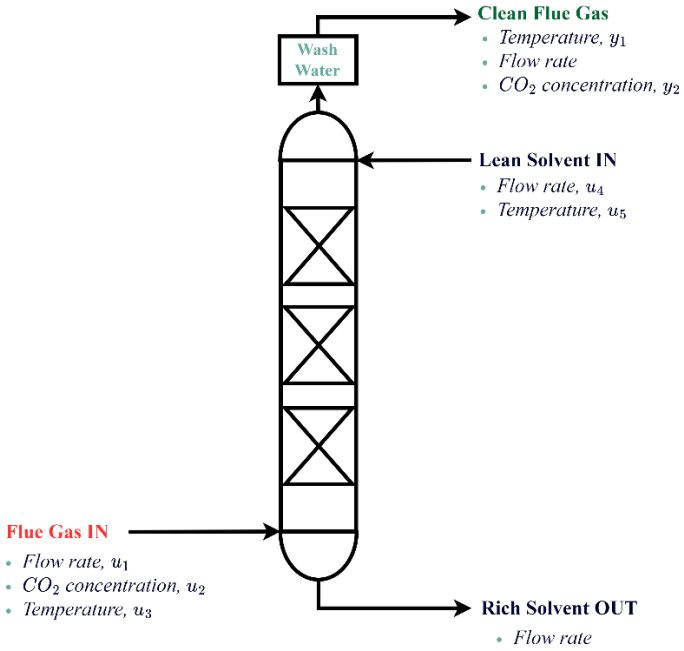


Fig. 3.8. Absorber unit for CO_2 capture

While data were collected by implementing multiple steps in each of the listed input variables [138,141], for model building in this work, the set of data considered is from step tests in the flue gas flowrate as shown in **Fig. B. 2** in **Appendix B**. Several challenges were faced while processing the pilot plant data for model building:

- Even though it was intended to introduce step changes only in the flue gas flowrate for the data presented in **Fig. B. 2**, other input variables in the pilot plant also kept changing although with much lower magnitude. This is because either those variables are disturbance variables such as flue gas CO_2 concentration and temperature or had some minor oscillations due to the plant control/measurement system. For model development, changes in all five input variables as observed in **Fig. B. 2** are considered. As the lean solvent inlet flow rate is observed to be significantly noisy, hence a k-nearest neighbor algorithm with a value of k=210 is used to remove the noise before using the data for model building (**Fig. B. 3** in **Appendix B** compares the lean solvent flow data before and after cleaning).
- Even though it was intended to implement a step change in the flue gas flowrate, actual implementation deviated to some extent from an exact step as observed in **Fig. B. 3** due to implementation error and control system in the pilot plant.
- As expected for the typical plant data of measured variables, quantitative estimate of the noise or its characterization is not available.

None of the SINDy algorithms could yield successful model building for the plant data for the multiple step changes in the flue gas flow rate. For the single step change in the flue gas flowrate, model building was done by using 83% of the data (226 out of 271 data points) while for the multiple step changes in the flue gas flowrate. 80% of the data (1681 out of 2101 data points) were used for model building.

Single step change in the flue gas flowrate

The plant data for the single step change in the flue gas flowrate from 6000 to 4300 lb/hr is considered. While three of the algorithms exhibit good performance for the entire prediction period as shown in Fig. 3.9 , models from the other four gave a poorer performance with three of them (BML bilinear I, SINDy A2 and SINDy B) requiring re-initialization to keep model results bounded in some regions from where the data was not used for training. Here 'reinitialization' means that instead of continuing the dynamic model simulation with the model values from the previous time step, output variables are set to the same values as the measured data at that time instant. The time instant when reinitialization is done is shown by the vertical lines in Fig. 3.11. It should be noted that even though the vertical lines are shown in Fig. 3.9, they are mainly shown for illustration indicating that no reinitialization was necessary at the same time instant when other three models are reinitialized.

Table 3.3 shows that the BML Bilinear algorithms and SINDy algorithms take considerably shorter time to build compared to the BIDSAM algorithms. BIDSAM I and II models have similar AICc and $\mathrm{AICc} + 2\mathrm{Cev}(\Sigma)$. Compared to those, BML Bilinear I model has much poorer performance, but the performance of BML Bilinear II model is considerably improved. Out of all the SINDy algorithms, SINDy A1 yields the best performance with its AICc being about 7% inferior to BIDSAM models. SINDy B is found to yield the worst performance with its AICc being about >1000% inferior compared to the BIDSAM algorithms. **Fig. B. 4** in **Appendix B** compare the performance of models of SINDy A1 and BIDSAM II algorithms when model trainings were done with and without cleaning the data.

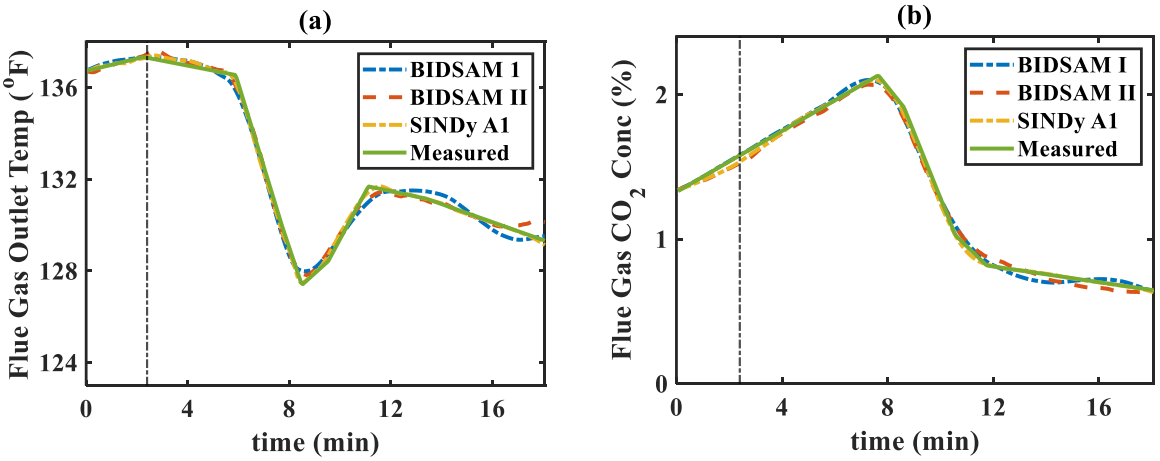


Fig. 3.9. Predictions of (a) flue gas outlet temperature (b) flue gas CO_2 concentration without re-initialization.

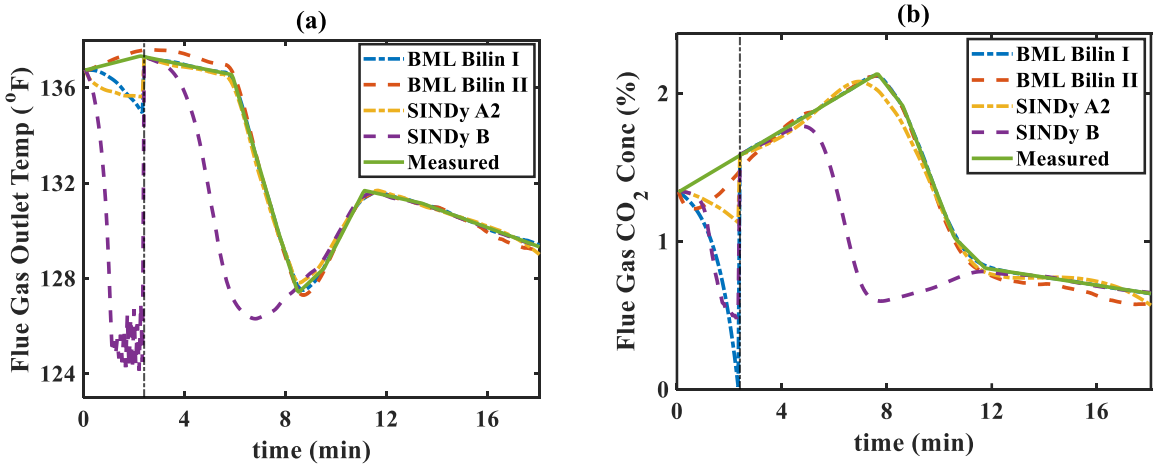


Fig. 3.10. Predictions of (a) flue gas outlet temperature (b) flue gas CO_2 concentration with re-initialization required.

Table 3.3. Performance indices of identified models for the single step change in the flue gas flow rate

Model	Model Building Time	No. of Parameters	% RMSE		$AICc + 2Cev(\Sigma)$	AIC_c
			T_{fg}	CO_2 Conc.		
Plant Measurement						
BIDSAM I	20 min	14	0.20	2.87	-363.8	-483.0
BIDSAM II	20 min	16	0.17	3.56	-350.9	-477.5
BML Bilinear I	19 sec	34	0.31	16.18	-	-12.9
BML Bilinear II	48 sec	34	0.17	7.00	-	-406.9
SINDy A1*	48 sec†	34	0.09	1.51	-	-448.0
SINDy A2*	9 sec†	33	0.34	6.55	-	-383.5
SINDy B*	11 sec†	23	3.26	32.96	-	5113

† – Model building time for SINDy without considering an average of 40mins spent for manual selection of appropriate class of basis functions and 'optimal' *threshold* values.

Multiple step changes in the flue gas flowrate

Kaheman et al. (2020) discussed the need to narrow down the library of basis functions defined for model building by using the SINDy algorithm. It was noted that consideration of basis functions that are not relevant to a system can lead to an increase in the condition number and consequently greater difficulty for the algorithm to select the optimal set. It has been shown [142] how the predictions of SINDy models can become unsatisfactory when tested with different input variables in the same range even when the model has been developed from datasets as large as 10,000 samples. For this case study, it was observed that different libraries of basis functions are required to be specified to effectively capture the dynamics of the system for multiple step changes

in multiple inputs. Furthermore, the SINDy algorithm is observed to be highly sensitive to the specified *threshold* value. For instance, if data from just a single step change in the flue gas flowrate from 5000 to 5700 lb/hr are considered, a combination of the Fourier and 2nd degree polynomial libraries with a *threshold* of 0.02 was needed while for the results presented in the earlier section (i.e., the flue gas flowrate changing from 6000 to 4300 lb/hr), only 2nd degree polynomial library was required with a *threshold* of 0.028. Overall, none of the SINDy algorithms could yield a single model for the entire transient data corresponding to the inputs given in Figure A3.1. For such data, the Ensemble-SINDy [143,144] which combines several SINDy models trained for the different input variable values is often used. Specifically, in the work of [142], 100 different models are developed for the different input values and a deep neural network is trained to approximate the final model predictions for a given set of inputs based on the ensembled SINDy models. [145,146] also demonstrated how ensemble SINDy can be employed to enhance the robustness of the SINDy algorithm to noise via subsampling, co-teaching and dropout methods. However, these approaches have only been tested on synthetic data and the multiple models resulting from the ensemble approach can lead to complex structures and possible loss of desired model interpretability. In addition, the co-teaching approach requires at least a simplified first-principles model to generate noise-free data to be used alongside with noisy measurements for the model development, but these simplified first-principles models may not be available for many systems or may give poor long-term predictions. Ensemble-SINDy is not evaluated in this work and therefore no results for the Ensemble-SINDy algorithms are presented for this case study. The single SINDy model developed for this data set yielded acceptable performance only for short-term predictions as shown in **Table B. 4** and **Fig. B. 4** and **Fig. B. 5** in **Appendix B.**

As described earlier, the model building approach of the BIDSAM algorithm does not require use of specific library of basis functions, rather it effectively selects the optimal subset from any defined general family of basis functions. Therefore, a single model could be built even when the dynamic data from multiple step changes are used. Fig. 3.11 shows the comparison between the measurements and results from all BIDSAM and BML Bilinear models. Table 3.4 compares key performance measures for the training and testing data. BIDSAM II is found to lead to the best results while the worst performance is obtained from BML Bilinear I. Results for the BIDSAM I is relatively similar to that from BIDSAM II with the result from BIDSAM I being only about 0.5% inferior for the training objective function.

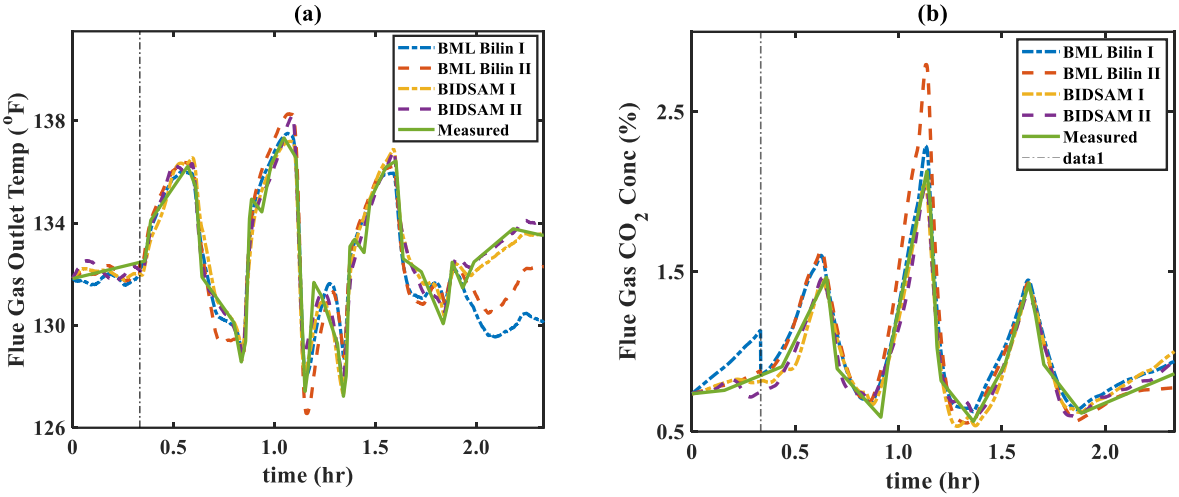


Fig. 3.11. Predictions of (a) flue gas outlet temperature (b) flue gas CO_2 concentration for series of step changes in flue gas flow rate.

Table 3.4. Performance indices of identified models for series of step changes in flue gas flow rate.

Model	Model Building Time	Number of Parameters	Training % RMSE		$AICc + 2Cev(\Sigma)$	Training % RMSE		$AICc$
			T_{fg}	CO_2 Conc.		T_{fg}	CO_2 Conc.	
BIDSAM I	27 min	28	0.43	8.93	-2648	0.39	8.72	-3440
BIDSAM II	33 min	40	0.35	7.98	-2662	0.34	7.72	-3552
BML Bilinear I	120 sec	34	1.00	10.20	-	1.08	12.15	-941
BML Bilinear II	97 sec	34	0.91	14.26	-	0.86	12.92	-1130

3.3.3 Hybridization with First-principles Model

In the preceding sections, the BIDSAM algorithm has been evaluated as a fully data-driven approach for both synthetic and real plant data. The focus of this section is on the hybridization of developed models using the proposed BIDSAM algorithm with first-principles model. Operational data from the superheater section of a coal-fired power plant boiler is considered. It is desired to accurately predict the main outlet steam temperature, $T_{st,main}$ as well as steam temperature at specific tube location, $T_{st,tubes}$ thereby providing a spatial resolution of the distribution of the steam temperature in the superheater system. This case study is particularly challenging due to the high level of noise and uncertainties in the data resulting from unstable ash conditions in the coal fired boiler.

Prediction of main outlet steam temperature

A high-fidelity model developed by another member of our research group [147] is used for the prediction of $T_{st,main}$. On validation of this model with the collected plant data, significant discrepancies were observed as shown in Fig. 3.12. Validation of first-principles model (a) prediction of Tst, main; (b) prediction error.

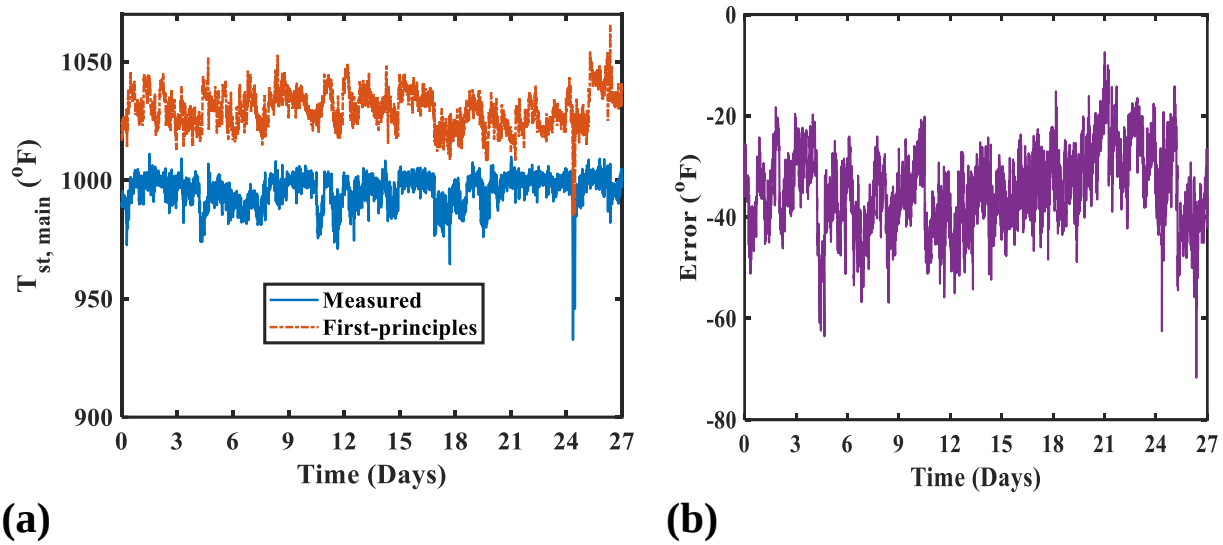


Fig. 3.12. Validation of first-principles model (a) prediction of $T_{st, main}$; (b) prediction error.

On analysis, the observed discrepancies are attributed to ash deposits on the boiler tube walls leading to inaccurate estimation of the heat transfer coefficient by the first-principles model. The BIDSAM algorithm is then employed to develop a steady state model that predicts a correction factor for the heat transfer coefficient in the prediction of average outlet steam temperature by the first-principles model. The input variables considered here are inlet flue gas flow rate and temperature, inlet steam flowrate, header valve positions, header spray flow rates and inlet steam temperatures at two specific locations. The hybrid model structure considered here is shown in Fig. 3.13 . Optimal corrective values for the heat transfer coefficients were obtained for a 12-day data with an optimizer, about 50% of the data was used for training a BIDSAM model for future predictions and then model was validated with 27 days data by hybridization with the first principles model.

Before hybridization, the standalone first-principles model predictions consistently varied from the measured plant data with up to 40^oF as shown in Fig. 3.12. However, hybridization with data-driven BIDSAM model (Fig. 3.14) kept the deviations in the range of $\pm 10^oF$ with a mean value of 0.4^oF.

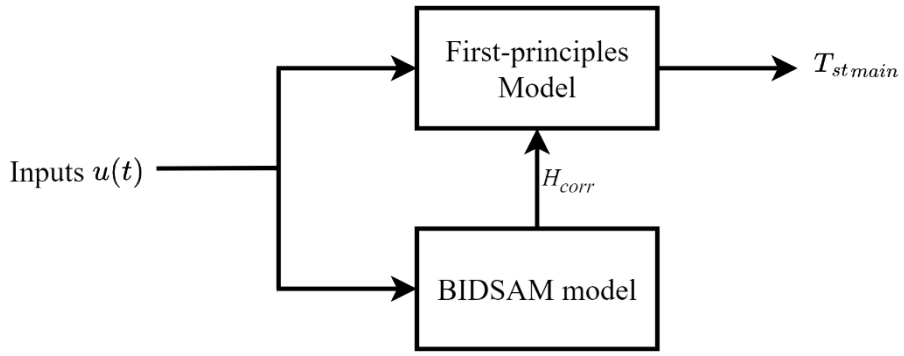


Fig. 3.13. Structure of hybrid model for heat transfer coefficient corrective term.

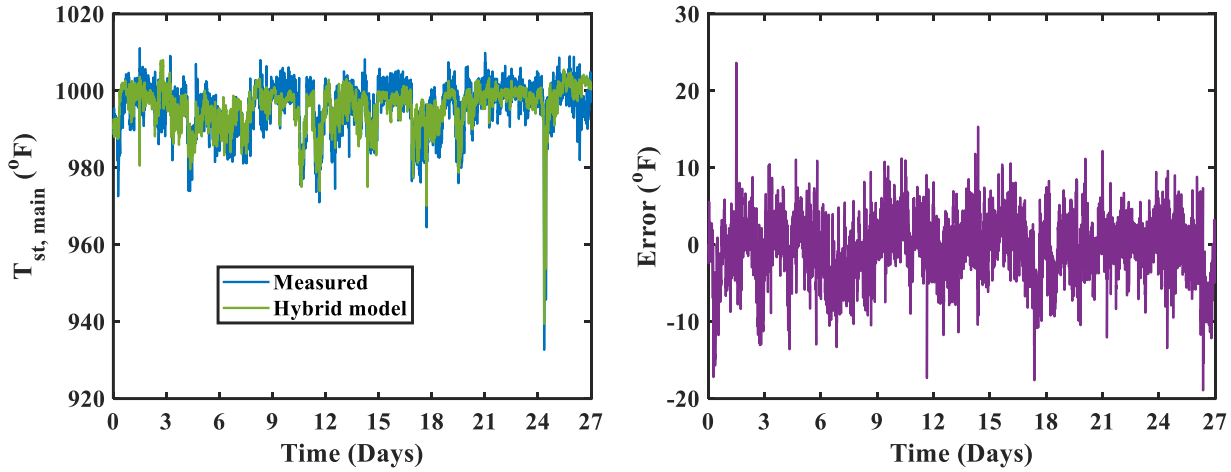


Fig. 3.14. Validation of hybrid model (a) prediction of $T_{st,\,main}$; (b) prediction error.

Prediction of tube steam temperatures

For prediction of steam temperatures in individual tubes at specific locations, a dynamic data-driven model is trained with the BIDSAM algorithm to model the dynamic discrepancies between temperatures at those specific locations and the main steam outlet temperature, $\Delta T_{st,tubes}$. The input variables remained the same as considered for the heat transfer coefficient corrective factor in but hybrid structure considered for this case is given by Fig. 3.15, Model training is done with a 16-hour dataset while validation is done for 37 days operational data. The validation plots shown in Fig. 3.16 shows the robustness of the developed algorithm as the RMSE values for the two locations shown are 7.8 and $11.9^o F$ despite the high level of noise in the data.

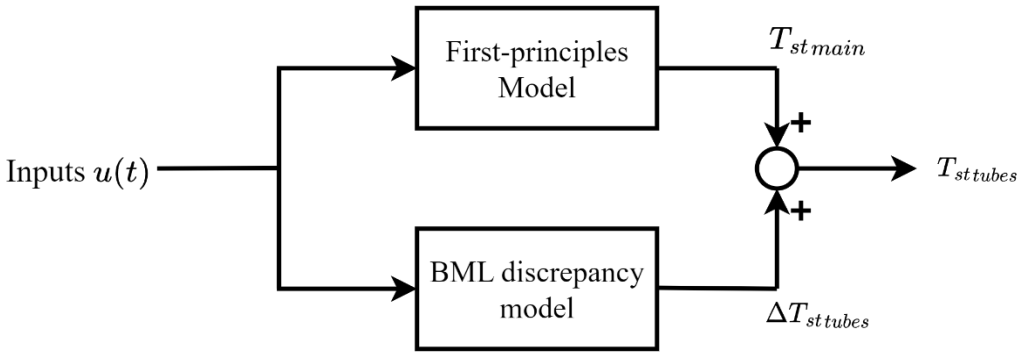


Fig. 3.15. Structure of hybrid model for predicting individual tube temperatures.

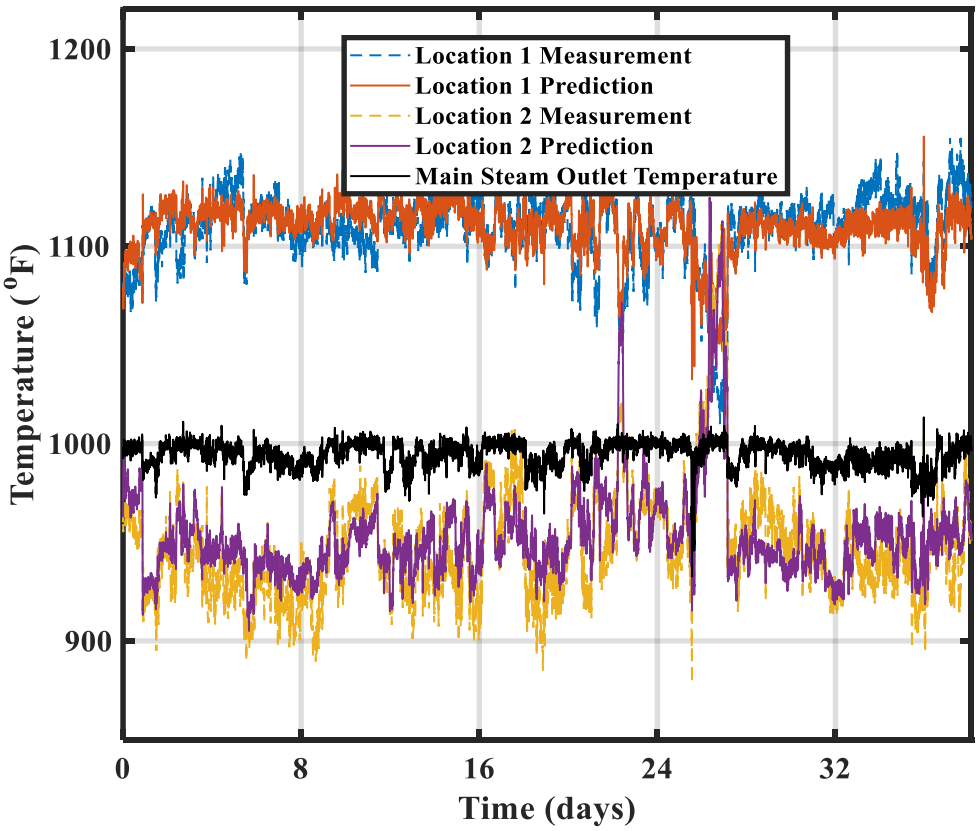


Fig. 3.16. Validation of hybrid model predicting steam temperatures at specific tube locations.

3.4 Mass Conservation by Sparse Data-Driven Models

Despite the superior performance exhibited by the BIDSAM algorithm compared to state-of-the-art algorithms, the resulting models are not guaranteed to conserve mass neither during training nor during simulation, hence may yield results that are not physically realistic. This also places a limitation on the extrapolation capabilities of the resulting models.

Meanwhile, for ANN models, the Physics-Informed Neural Network (PINN) has been proposed to address the challenge of lack of model interpretability and limited extrapolation capabilities through the resulting Physics-informed machine learning models[75,148–151]. This approach incorporates physical laws while training neural networks by including model equations for physical/chemical mechanisms, symmetry constraints, and/or intuitive physical constraints as components of ANNs [152,153]. In addition to achieving interpretability, the incorporated physical laws can also help to attain plausible predictions even when limited data are available for training. However, the implementation of PINNs often requires a significant knowledge of the system such as specific transport mechanisms or thermodynamic relations for the system that is being modeled. Hence, these approaches are often highly system specific, and can consume considerable resources to develop.

For satisfying physics constraints under the Bayesian framework, several approaches have been developed. Bayesian Physics-Informed Neural Network (B-PINN)[154] has been developed to simultaneously enhance model robustness by quantifying model uncertainties while also imposing physics constraints during model development. A parameterized surrogate model in the form of Bayesian Neural Network (BNN) [155–157] is combined with a suitable approach for estimating posterior distribution of the model parameters. BNN seeks to exploit the probabilistic nature of Bayesian approach in quantifying uncertainties through probabilistic representation of a priori knowledge, domain specific knowledge, model selection schemes, parameter estimation methods and noise estimation techniques[158]. Algorithms for estimating the posterior weight distribution include use of Gaussian approximation[155,157], that avoids Gaussian approximation by using Hybrid Monte Carlo method[156], use of the expectation maximization (EM) algorithm with (Extended) Kalman Filtering for estimating parameters of a fixed model structure based on some knowledge of the covariance of states/measurements[159]. While B-PINNs have been shown to be more robust than PINNs, the approach inherits the same limitation of requiring specific first-principles model equations for their implementation. Besides, mass conservation is not exactly satisfied since the PINN approach simply brings in the available physical knowledge as regularization terms in the loss function[151–153]. In addition, the approach has been noted to be faced with some other limitations such as conflicting results when different loss terms are used and difficulty/inefficiency in representing complex systems using single multi-layered perception (MLP) networks[152]. This is similar to the approaches followed for developing/training the Physics Constrained Neural Networks (PCNN)[153,160] where custom networks are designed to enforce initial/boundary conditions while including the model equations given by partial differential equations (PDEs), ordinary differential equation (ODEs), etc., in the loss function.

In summary, it is our observation that the existing approaches for data-driven model building do not necessarily consider exact satisfaction of mass. Conservation of mass during training and simulation of chemical engineering systems is critical for generating meaningful and physically realistic results. In a recent work [161], an algorithm was presented for ANNs that can exactly satisfy mass. However, to the best of our knowledge, there is no existing work on sparse data-driven model building (both steady-state and dynamic) using candidate basis functions for exactly satisfying conservation of mass. The objective of this work is to develop mass-constrained

BIDSAM (MCBIDSAM) by extending our previously developed BIDSAM algorithm for exactly satisfying mass conservation during both training and simulation. In the discussion below, for simplicity, we refer to the model developed by the BIDSAM and MCBIDSAM algorithms as BIDSAM model and MCBIDSAM model, respectively. We propose two algorithms to achieve this objective. In the first approach, the BIDSAM algorithm that does not necessarily satisfy mass conservation is augmented with a data reconciliation step. In the second approach, the unique structure of the BIDSAM model is exploited to formulate a set of equality constraints that guarantee mass conservation. Both algorithms use only boundary conditions to guarantee mass conservation which, we believe, can be constructed for many chemical engineering systems with reasonable effort. Exact satisfaction of mass constraints by the dynamic model can be difficult since mass conservation constraints can be applied only when the steady-state data are considered or if the time-varying system holdup information is available. While measurement of holdup can be challenging for distributed systems, it can be available for lumped or well mixed systems such as CSTRs. Algorithms are extended based on whether the holdup information is available.
In this section, we discuss the two MCBIDSAM algorithms that are implemented on models obtained from the BIDSAM algorithm discussed in Section 2.3.2 to ensure the conservation of mass during training and predictions.

3.4.1 Algorithm I

This first approach is shown in in **Fig. 3.17** (a) where the outputs from the BIDSAM model are propagated through a dynamic data reconciliation (DDR) block for conserving mass. DDR involves solution of an optimization problem (QP/NLP). The BIDSAM algorithm also includes solution of a two-stage optimization problem (MINLP). For solving this overall problem, we develop a tearing approach as shown in **Fig. 3.17** (b). This approach enables solving the BIDSAM algorithm and DDR separately and an outer loop for the convergence of the overall problem. While the problem can be solved monolithically using a MINLP solver, the tearing approach enables use of the BIDSAM algorithm developed in Section 3.3, while the DDR problem is solved by an efficient QP solver (as mass conservation constraints are often linear) or general NLP solver.

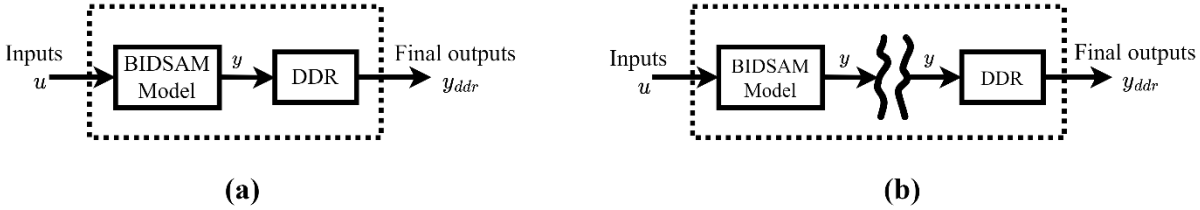


Fig. 3.17. (a) Block diagram for the MCBIDSAM model development using Algorithm I; (b) Tearing Approach for Algorithm I.

The optimization problem for the DDR block is given by:

$$\min_{\Omega, y_{ddr}} \sum_i (\Omega y^k - y_{ddr})^T W_1^{-1} (\Omega y^k - y_{ddr}) + \sum_i (y_{data} - y_{ddr})^T W_2^{-1} (y_{data} - y_{ddr}) \quad (3.37)$$

$$\boldsymbol{s.t}\ \text{Mass conservation laws}$$

In Eq. (3.37), y denotes the output of the BIDSAM model, Ω denotes a linear operator that is used as a degree of freedom (DOF), y_{data} denotes the measurements, y_{ddr} represents the final model outputs, W_1 and W_2 are the weighting matrices. Mass conservation laws are enforced by using atom balance constraints in the following form:

$$y_{j\ acc,k+1} = \sum n_{j,p_i}(u_{p,k}) - \sum n_{j,q_o}(y_{q,ddr,k+1}) \tag{3.38}$$

where $y_{j\ acc}$ is the accumulation of the atom j in the system, n_{j,p_i} is the number of atoms of element j for input variables p while n_{j,q_o} are the number of atoms of element j for output variables q. For steady-state model building, the left-hand side of Eq. (3.38) becomes zero. For dynamic model building if the holdup measurements are available, then Eq. (3.38) is considered for all time steps. If the holdup information is not available, then Eq. (3.38) is simply applied to the steady-state data and the DDR block is omitted for the transient data. We define steady state zones as periods in which the relative changes in the values of all measured outputs are maintained within a specified bandwidth, e.g. $\pm 6\%$ while all input variables remain constant similar to what is done in a previous work [161]. Unlike what is done by [161] where two networks were developed on the assumption that the time series data for the system considered can be distinctly partitioned into steady-state and dynamic zones, in this work, a single MCBIDSAM model is developed that reduces to the BIDSAM model for transient data zones when holdup information is not available for dynamic system.

MCBIDSAM Algorithm I for solving the inverse problem is shown in Fig. 3.18. Sparse models are built by using the BIDSAM algorithm described in Section 3.2. This begins with the construction of library of basis functions followed by initial ranking of individual basis functions based on their respective optimal value of the objective function given in Section 3.2. This is followed by executing the previously developed B&B algorithm as detailed in Section 3.3. The candidate models obtained are ranked based on the values of the objective function given in Eq.(3.26). Following this, the N_{top} top rank candidate models are considered for enforcement of mass constraint by the DDR block. It should be noted that N_{top} is much higher than the number of models that are finally selected so even though the relative ranking of these candidate models might change upon enforcing mass constraints, the final models are still subset of these candidate models. This step avoids enforcing mass constraints to millions of candidate models that are not likely to be selected thus reducing computational expense of the EM algorithm considerably and avoiding infeasibility of satisfaction of mass constraints by many candidate models. The optimization problem in Eq. (3.37) is solved to obtain parameter Ω and final model outputs y_{ddr} that satisfy mass balance. During subsequent iterations, Ω and y values from previous iteration are used as initial guess for MAP estimate of the model parameters using Bayesian inferencing and the EM algorithm. The iteration continues until convergence, i.e., till the Euclidean norm of the changes in the Ω values satisfies a tolerance.

For the forward (simulation) problem, the optimization problem for the DDR block is given as follows:

$$\min_{y_{ddr}} \sum_i (\Omega y - y_{ddr})^T W_1^{-1} (\Omega y - y_{ddr}) \tag{3.39}$$
$$s.t \text{ Mass balance}$$

where optimal values of $\boldsymbol{\Omega}$ that are obtained from solving the MCBIDSAM problem are used. For both the forward and inverse problems, the Interior Point OPTimization [162] (IPOPT) is used for solving the equality constrained optimization problem for the DDR.

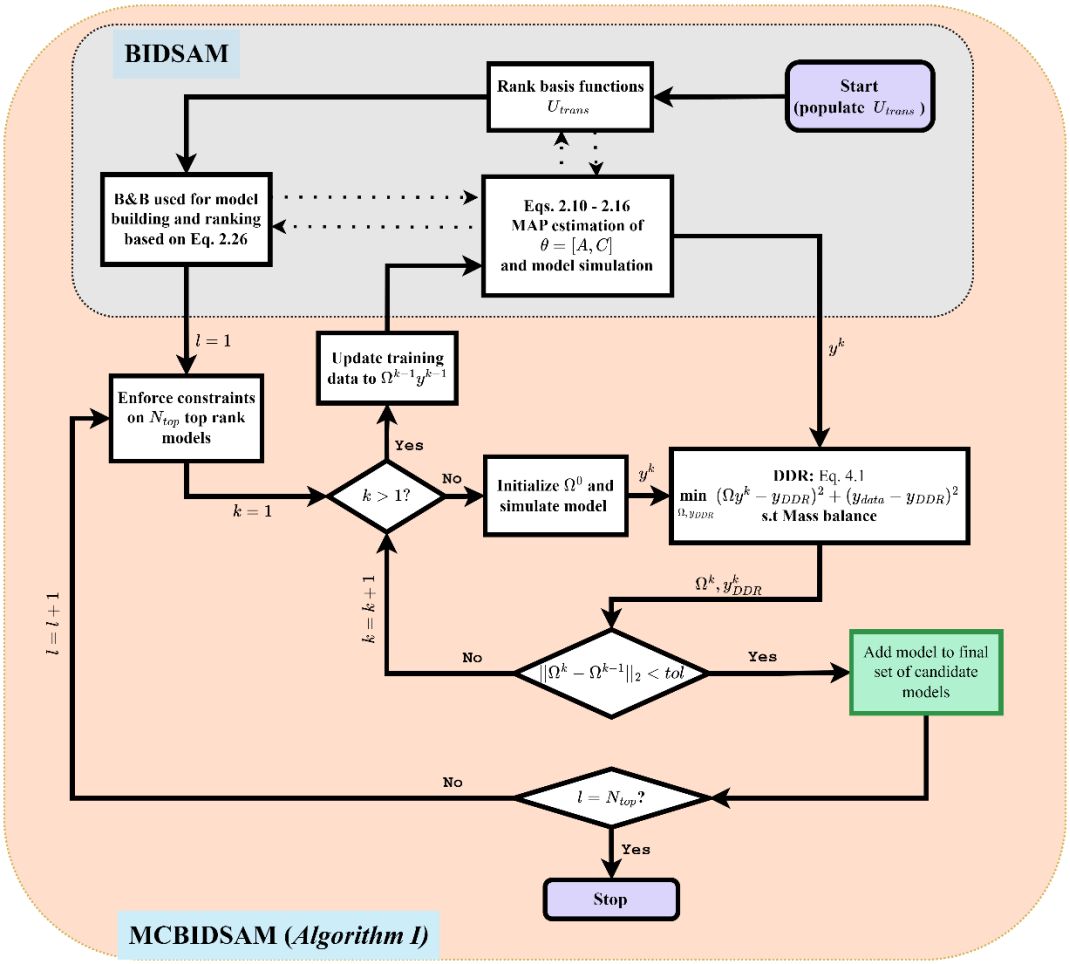


Fig. 3.18. Algorithm I for solving inverse problem for MCBIDSAM model.

3.4.2 Algorithm II

MCBIDSAM Algorithm II is presented in **Fig. 3.19**. This algorithm seeks to exploit the advantage of the structure of the BIDSAM models that are linear in parameter to derive a set of equality constraints on the model parameters guaranteeing the satisfaction of the mass conservation rules. Similar to *Algorithm I,* the BIDSAM algorithm is first used to obtain a set of N_{top} hierarchically ranked sparse models. Then a set of equality constraints are derived based on the identified model, after which an optimization problem is solved to enforce the constraints. Detailed description of the method for deriving the set of equality constraints for exact satisfaction of mass conservation rules is discussed below for steady state and dynamic model building.

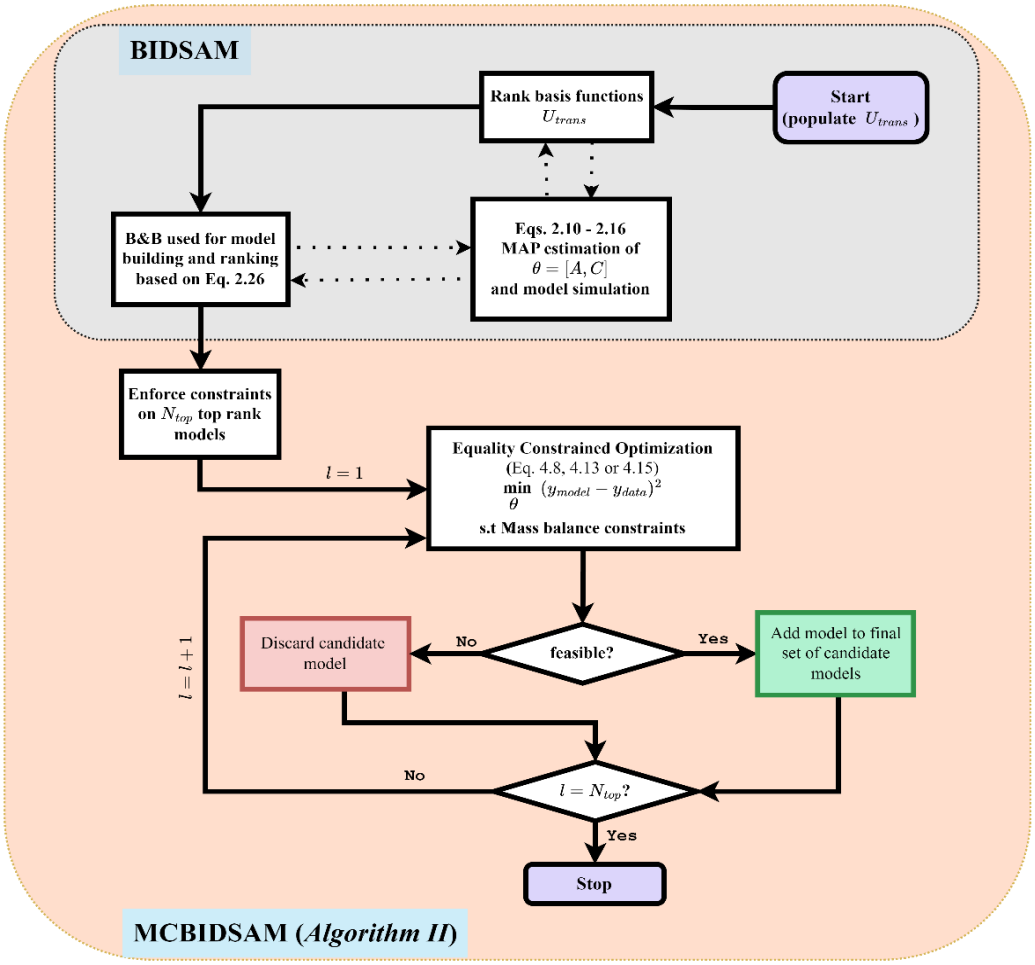


Fig. 3.19. Algorithm II for solving inverse problem for MCBIDSAM model.

Steady state model building

For development of the general approach, let us consider a system with two input variables and four measured output variables as shown in Fig. 3.20. Also consider a simple candidate model with a set of three basis functions obtained using the BIDSAM algorithm given by Eq.(3.40) where u^*_{trans} denotes any general set of nonlinear transformation of the input variables and their interactive effects with the output variables.

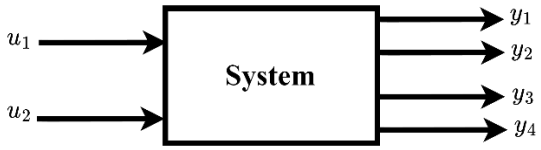


Fig. 3.20. Block diagram of a system with 2 inputs and 4 outputs.

$$\begin{pmatrix} y_{1,k} \\ y_{2,k} \\ y_{3,k} \\ y_{4,k} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{pmatrix} \begin{pmatrix} u_{1,k} \\ u_{2,k} \\ u^*_{trans} \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} \tag{3.40}$$

Under steady state conditions, the accumulation term in Eq.(3.38) becomes zero. Suppose u_1 and u_2 represent flowrates of two feed streams to the system, y_1, y_2, and y_3 represent flowrates of product streams out of the system, and y_4 represent any other measured output variable other than product flowrate (e.g., temperature). For atom balance carried out with respect to atom j, the right-hand side of Eq.(3.38) for the system in Fig. 3.20 can be written as follows:

$$n_{j,1_i}u_{1,k} + n_{j,2_i}u_{2,k} - n_{j,1_o}y_{1,k} - n_{j,2_o}y_{2,k} - n_{j,3_o}y_{3,k} = 0 \tag{3.41}$$

Note that since y_4 is not related to product flowrate, $n_{j,4_o} = 0$ and hence y_4 does not appear in Eq.(3.41). Similar rule applies to transformations of input variables, u^*_{trans} and input variables that are not direct measurements of species flowrates into the system. On substituting the expressions for $y_i's$ in Eq. (3.40) into Eq.(3.41), we obtain:

$$\left(n_{j,1_i} - \sum_{q=1}^{3} n_{j,q_o}a_{q1}\right)\boldsymbol{u_{1,k}} + \left(n_{j,2_i} - \sum_{q=1}^{3} n_{j,q_o}a_{q2}\right)\boldsymbol{u_{2,k}} - \sum_{q=1}^{3} n_{j,q_o}a_{q3}\,\boldsymbol{u^*_{trans}} - \sum_{q=1}^{3} n_{j,q_o}d_q = 0 \tag{3.42}$$

As u_1, u_2, u^*_{trans} and d_q are independent and Eq. (3.42) must hold true for all values of u_1, u_2, u^*_{trans} and d_q, we have that:

$$\begin{gathered} n_{j,p_i} - \textstyle\sum_{q=1}^{3} n_{j,q_o}a_{qp} = 0 \quad \text{for } p = 1,2 \\ \textstyle\sum_{q=1}^{3} n_{j,q_o}a_{q3} = 0\ ,\ \sum_{q=1}^{3} n_{j,q_o}d_q = 0 \end{gathered} \tag{3.43}$$

Satisfying these set of equality constraints on the model parameters will guarantee that model predictions always satisfy mass conservation requirements as defined by Eq.(3.40) regardless of the values of the input variables or the training data set for this model structure. The following optimization problem is solved using IPOPT[162] to obtain the optimal set of model parameters satisfying these constraints:

$$\begin{aligned} &\min_{\theta=\{\boldsymbol{a_{qp}},\boldsymbol{d_q}\}} \sum (y_{model} - y_{data})^2 \\ &\boldsymbol{st} \quad n_{j,p_i} - \textstyle\sum_{q=1}^{3} n_{j,q_o}a_{qp} = 0 \quad \textit{for p=1,2} \\ &\textstyle\sum_{q=1}^{3} n_{j,q_o}a_{q3} = 0 \\ &\textstyle\sum_{q=1}^{3} n_{j,q_o}d_q = 0 \end{aligned} \tag{3.44}$$

Although this formulation has been demonstration for a system with two input and three output variables, the approach is applicable to any number of input and output variables. However, it is important to note that this problem set up requires that the flowrates of all mass input and output streams, directly or indirectly, are included as inputs and outputs, respectively and that the model includes linear basis functions of all mass inputs. We believe that this is the case for model building of many chemical systems and thus is not very restrictive. Other than these requirements, the approach developed above holds for satisfying mass conservation regardless of the set of basis function(s) selected by the BIDSAM algorithm for the system. It could be noted, for example, that in the above derivation, $\boldsymbol{u^*_{trans}}$ is used to denote any form of nonlinear transformation of any input variable including their interactive effects with the measured output variables, but is not included in formulating the mass conservation constraints.

Dynamic model building

For dynamic model building of the system in Fig. 3.20, Eq.(3.40) is modified for a 2 input, 4 output system as:

$$\begin{pmatrix} y_{1,k+1} \\ y_{2,k+1} \\ y_{3,k+1} \\ y_{4,k+1} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} y_{1,k} \\ y_{2,k} \\ y_{3,k} \\ y_{4,k} \end{pmatrix} + \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ c_{41} & c_{42} & c_{43} \end{pmatrix} \begin{pmatrix} u_{1,k} \\ u_{2,k} \\ u^*_{trans} \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} \quad (3.45)$$

The exact satisfaction of mass conservation in this case can be achieved either for both steady state and transient conditions, or only for steady state conditions depending on the availability of mass holdup (accumulation) measurements of the chemical species in the system that is being evaluated.

Mass conservation at steady state and during transient condition

Two different possible scenarios are demonstrated. In a very general case where the holdup of each chemical species $y_{q\ acc}$ is known, the accumulation of the atom j in the system, $y_{j\ acc}$ in Eq. (3.38) can be written as follows to satisfy the atom balance:

$$\sum n_{j,q_o}(y_{q,acc,k+1}) = \sum n_{j,p_i}(u_{p,k}) - \sum n_{j,q_o}(y_{q,ddr,k+1}) \quad (3.46)$$

Depending on the form of expressions representing $y_{q\ acc}$, Eq. (3.46) can be simplified in a similar way as Eq.(3.42) to obtain the set of constraints for the optimization problem.

Mass conservation only at steady state

In situations where holdup information is not available and/or there is significant difficulty in the development of accurate simple models for $y_{q,acc}$, mass conservation can still be guaranteed under steady state conditions. At steady state where $y_{i,k+1} = y_{i,k}$, the expression for y_1 in Eq.(3.45) simplifies to

$$(1 - a_{11})y_{1,k} = a_{12}y_{2,k} + a_{13}y_{3,k} + c_{11}u_{1,k} + c_{12}u_{2,k} + c_{13}\,u^*_{trans} + d_1 \quad (3.47)$$

We substitute Eq.(3.47) and similar expressions for $y_{2,k}$ and $y_{3,k}$ into Eq.(3.41) and on simplification (see **Appendix B**) and following the approach similar to steady-state model building, we obtain the following set of equality constraints:

$$\begin{aligned}
&\frac{n_{c,2_o}c_{21}}{1-a_{22}} + \frac{n_{c,3_o}c_{31}}{1-a_{33}} = 0 \\
&\frac{n_{c,1_o}c_{12}}{1-a_{11}} + \frac{n_{c,3_o}c_{32}}{1-a_{33}} = 0 \\
&\frac{n_{c,1_o}c_{13}}{1-a_{11}} + \frac{n_{c,2_o}c_{23}}{1-a_{22}} = 0 \\
&\frac{n_{c,1_o}c_{11}}{1-a_{11}} + \frac{n_{c,2_o}c_{21}}{1-a_{22}} + \frac{n_{c,3_o}c_{31}}{1-a_{33}} - n_{c,1_i} = 0 \\
&\frac{n_{c,1_o}c_{12}}{1-a_{11}} + \frac{n_{c,2_o}c_{22}}{1-a_{22}} + \frac{n_{c,3_o}c_{32}}{1-a_{33}} - n_{c,1_i} = 0 \\
&\frac{n_{c,1_o}c_{13}}{1-a_{11}} + \frac{n_{c,2_o}c_{23}}{1-a_{22}} + \frac{n_{c,3_o}3}{1-a_{33}} = 0 \\
&n_{c,1_o}d_1 + n_{c,2_o}d_2 + n_{c,3_o}d_3 = 0
\end{aligned} \quad (3.48)$$

The BIDSAM model parameters are then optimized by solving the following equality constrained optimization problem to guarantee the satisfaction of mass balance by all predictions of the model under steady state conditions:

$$\min_{\theta=\{\boldsymbol{a_{qp}},\boldsymbol{d_q}\}} \sum (y_{model} - y_{data})^2 \tag{3.49}$$

$$\begin{aligned}
\textbf{\textit{s.t}} \quad & \frac{n_{c,2_o} c_{21}}{1-a_{22}} + \frac{n_{c,3_o} c_{31}}{1-a_{33}} = 0 \\
& \frac{n_{c,1_o} c_{12}}{1-a_{11}} + \frac{n_{c,3_o} c_{32}}{1-a_{33}} = 0 \\
& \frac{n_{c,1_o} c_{13}}{1-a_{11}} + \frac{n_{c,2_o} c_{23}}{1-a_{22}} = 0 \\
& \frac{n_{c,1_o} c_{11}}{1-a_{11}} + \frac{n_{c,2_o} c_{21}}{1-a_{22}} + \frac{n_{c,3_o} c_{31}}{1-a_{33}} - n_{c,1_i} = 0 \\
& \frac{n_{c,1_o} c_{12}}{1-a_{11}} + \frac{n_{c,2_o} c_{22}}{1-a_{22}} + \frac{n_{c,3_o} c_{32}}{1-a_{33}} - n_{c,1_i} = 0 \\
& \frac{n_{c,1_o} c_{13}}{1-a_{11}} + \frac{n_{c,2_o} c_{23}}{1-a_{22}} + \frac{n_{c,3_o} 3}{1-a_{33}} = 0 \\
& n_{c,1_o} d_1 + n_{c,2_o} d_2 + n_{c,3_o} d_3 = 0
\end{aligned}$$

<u>Mass conservation at steady state with error minimization during transience</u>

While the above formulation for dynamic model guarantees the conservation of mass at steady state conditions, significant violations of the same may occur during transience. Also, even when holdup information is available, the development of model for $y_{q,acc}$ and simplification of Eq.(3.46) can easily become complicated due to different sets of basis functions that may be required to model the species accumulation compared to those used for predicting the species concentrations. Hence, we consider an alternative option for using additional information about the species holdup in the system, if available, directly or indirectly, to improve model performance. In this case, we derive the equation for the prediction of species holdup in the system by the same set of parameters that predict the species concentrations/flowrates thereby constraining the predictions to satisfy mass balance. Updating Eq.(3.38) and substituting the equations for $y_{i,k+1}$ (for $i = 1,2,3$) given by Eq.(3.48) into Eq.(3.38) and upon simplification, we obtain the following:

$$\begin{aligned}
y_{j\ acc\ k+1} = & -\begin{pmatrix} \sum_{q=1}^{3} n_{j,q_o} a_{q1} \\ \sum_{q=1}^{3} n_{j,q_o} a_{q2} \\ \sum_{q=1}^{3} n_{j,q_o} a_{q3} \end{pmatrix}^T \begin{pmatrix} y_{1,k} \\ y_{2,k} \\ y_{3,k} \end{pmatrix} + \begin{pmatrix} n_{j,1_i} - \sum_{q=1}^{3} n_{j,q_o} c_{q1} \\ n_{j,2_i} - \sum_{q=1}^{3} n_{j,q_o} c_{q2} \\ -\sum_{q=1}^{3} n_{j,q_o} c_{q3} \end{pmatrix}^T \begin{pmatrix} u_{1,k} \\ u_{2,k} \\ u^*_{trans} \end{pmatrix} \\
& + \sum_{q=1}^{3} n_{j,q_o} d_q
\end{aligned} \tag{3.50}$$

Eq.(3.50) which predicts species accumulation based on the same model parameters predicting the species concentrations has taken the same form of the expressions in Eq.(3.45) in this case. However, this can take some other forms with new sets of basis functions depending on other forms of Eq.(3.45). Derivations of other possible forms such as for the specific case of an isothermal reactor are given in **Appendix B**.

Since Eq.(3.50) is derived from conservation equation, the greater its accuracy in predicting the actual species holdup in the system, the closer the original BIDSAM model is to exact satisfaction of mass conservation rule. Hence, considering the same set of equality constraints given in Eq.(3.48), the model parameters can be optimized with the modified objective function given by the following expression:

$$\min_{\theta=\{\boldsymbol{a_{qp}},\boldsymbol{d_q}\}} \sum (y_{model} - y_{data})^2 + \Psi \sum \left(y_{j\,acc_{model}} - y_{j\,acc_{data}}\right)^2 \tag{3.51}$$

where Ψ is a parameter introduced to adjust the weighting of the two components of the objective function since they may be of different orders. For the results shown in this work, the objective function is calculated from the normalized values of each variable and Ψ is set to a value of 1 denoting equal weighting.

3.5 Case Studies

Two case studies are considered for the evaluation of the proposed algorithms: the absorber unit of a post combustion CO_2 capture plant and an isothermal CSTR reactor. In each case, both steady state and dynamic models are built both from *true* data and noisy data with constant bias. The $RMSE$ criteria given by Eq.(3.52) is used for comparing the performances of the models obtained from the different algorithms:

$$RMSE = \sqrt{\frac{1}{k_n} \times \sum_{k=1}^{k_n} \left(y_k - h(u_{k-1}, y_{k-1}, \theta)\right)^2} \tag{3.52}$$

where k_n is the total number of time steps considered. In the first case study, the training data is simulated from an aspen dynamic model detailed in [138], while the second case study is simulated from the ordinary differential equations modeling the system.

3.5.1 Absorber unit for CO_2 capture plant

The absorber unit of a solvent-based post combustion CO_2 capture unit[138,139,141,163] discussed in Section 3.3 is considered. Here we consider the simulated data from Aspen dynamic model of the plant to which simulated noise is added. Unlike real plant data, the simulated data provides us with the knowledge of the *truth* to which we can compare the mass constrained model for correctness of its predictions while satisfying mass balance. The schematic diagram for the decoupled absorber unit with variables of interest considered here is shown in Fig. 3.21.

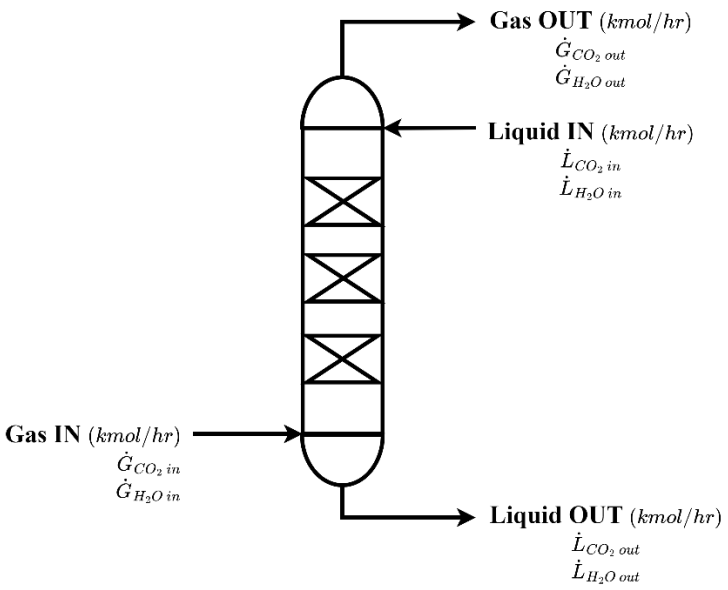


Fig. 3.21. Schematic of CO_2 absorber unit

The changing variables considered for model development are the molar flowrates of CO_2 and H_2O in the gas and liquid inlet streams $\left(\dot{G}_{CO_2\ in}, \dot{G}_{H_2O\ in}, \dot{L}_{CO_2\ in}\ and\ \dot{L}_{H_2O\ in}\right)$ while the measured output variables are molar flowrates at the outlet streams $\left(\dot{G}_{CO_2\ out}, \dot{G}_{H_2O\ out}, \dot{L}_{CO_2\ out}\ and\ \dot{L}_{H_2O\ out}\right)$. The algorithms proposed in this work are tested both for steady state and dynamic data simulated for this system and the results are presented below.

3.5.2 Results for steady state model

Fig. 3.22 show the prediction of $\dot{L}_{H_2O\ out}$ by a model developed from noisy data with bias using *Algorithm II*. Fig. 3.22 (a-b) show how the BIDSAM model without mass constraints fits the measurements but deviates from the truth while Fig. 3.22 (c-d) show how MCBIDSAM model gives predictions close to the truth while satisfying mass conservation even though it appears to deviate from the measured values. **Appendix B** shows similar plots for the prediction of $\dot{G}_{CO_2\ out}$ by the same model and $\dot{L}_{CO_2\ out}$ by the model from *Algorithm I*. All plots indicate that the proposed algorithms result in models with minimum bias from the truth.

Table 3.5 summarizes the results for models developed from both *true* and noisy data with bias using *Algorithms I* and *II*.

Table 3.5. Performance measures of the steady state models of the absorber unit of CO_2 capture plant.

Algorithm	**Mass constrained?**	**No. of parameters**	**RMSE (*kmol/hr*)**			
			$\dot{G}_{CO_2\ out}$	$\dot{G}_{H_2O\ out}$	$\dot{L}_{CO_2\ out}$	$\dot{L}_{H_2O\ out}$
Trained using *True* Data						
Algorithm I	No	12	3e-4	0.02	0.02	0.43
	Yes	16	3e-4	0.06	3e-4	0.06
Algorithm II	No	24	3e-4	0.01	0.01	0.38
	Yes		2e-4	5e-3	2e-4	5e-3
Trained using Noisy Data with Bias						
Algorithm I	No	12	9e-4	0.09	0.13	2.14
	Yes	16	7e-4	0.08	7e-4	0.08
Algorithm II	No	24	9e-4	0.09	0.13	2.15
	Yes		8e-4	0.09	8e-4	0.09

Since the data used for training involves mainly changes in the liquid flowrates, the gas concentrations remain largely unchanged. Both *Algorithms I* and *II* exhibit similar performance by reducing the largest $RMSE$ values in $\dot{L}_{H_2O\ out}$ predictions by the MCBIDSAM models from 2.14 *kmol/hr* and 2.15 *kmol/hr* to 0.08 *kmol/hr* and 0.09 *kmol/hr*, respectively. As for the model size, *Algorithm I* is observed to give the sparsest model in this case with only 67% number of parameters required by *Algorithm II* to achieve similar level of accuracy. This is because *Algorithm I* does not necessarily require the linear terms of the input variables to be present for its implementation as it is the case for *Algorithm II*.

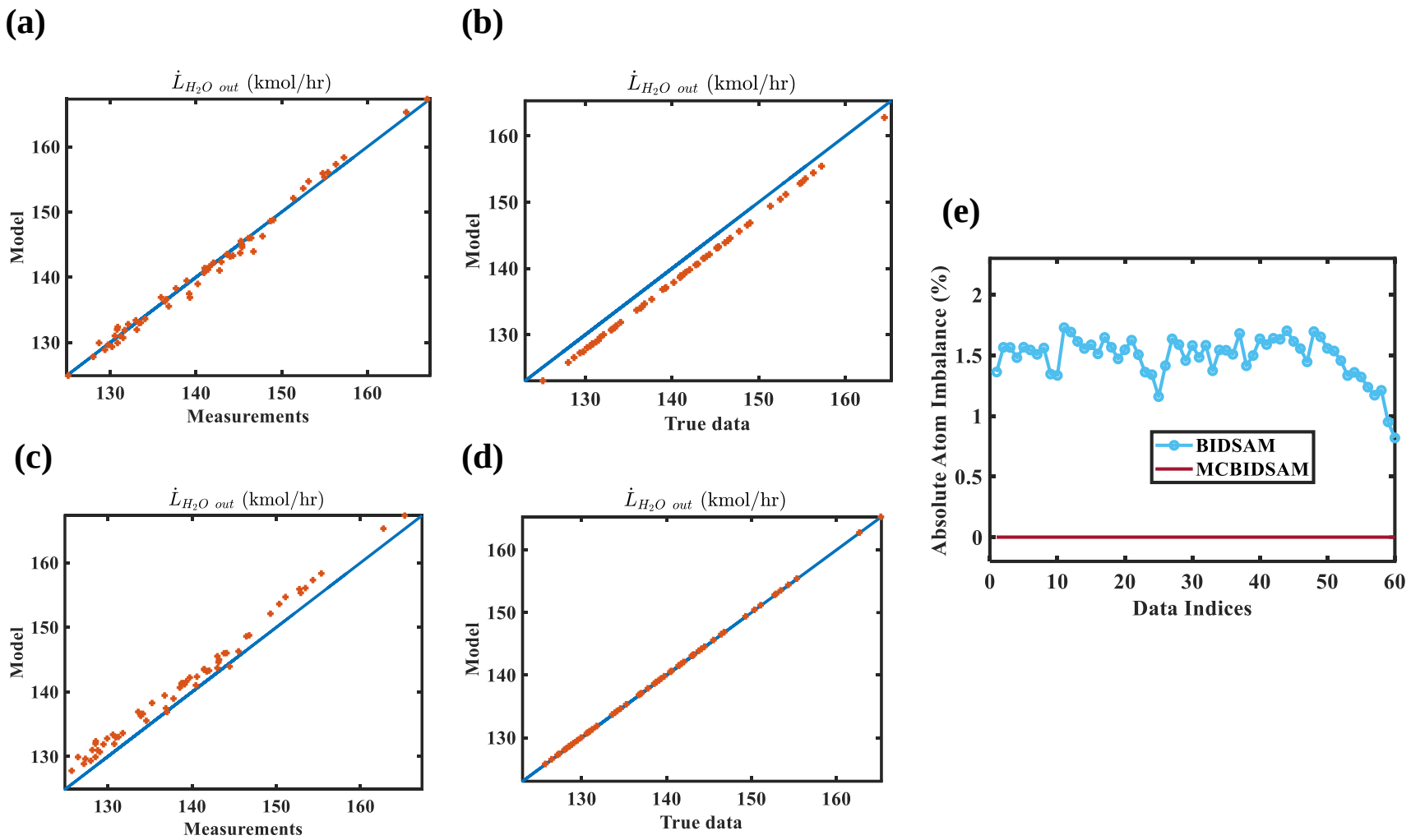
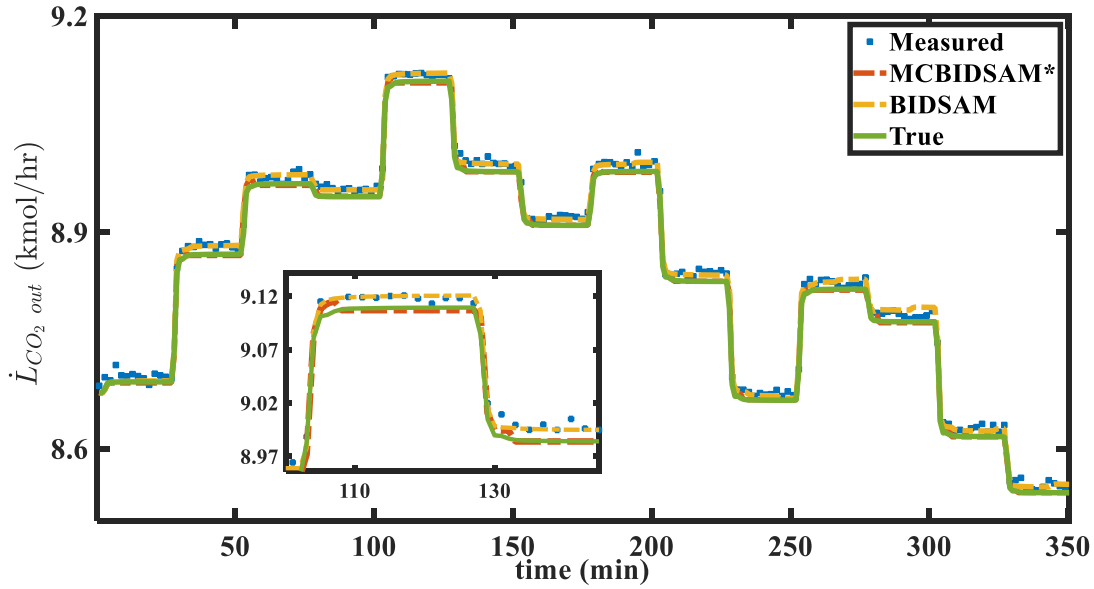


Fig. 3.22. Validation plots for model developed from noisy data using Algorithm II: (a)-(b) BIDSAM, (c)-(d) MCBIDSAM, (e) absolute percentage error in atom balance.

3.5.3 Results for dynamic model

In this case, we consider the optimization problem in Eq.(3.48) for the satisfaction of mass conservation at steady state since we do not have the species hold up information. The validation plots for prediction of $\dot{L}_{CO_2\ out}$ by models developed from noisy data with bias using *Algorithms I* and *II* are shown in Fig. 3.23. It was observed that for the unconstrained scenario, *Algorithm I* model performed better with maximum $RMSE$ value of 0.17 *kmol/hr* compared to 0.25 *kmol/hr* obtained using *Algorithm II* that also has about 30% more number of parameters. Again, this is because *Algorithm I* does not require the selection of the linear basis functions as for *Algorithm II*. Hence the top model for *Algorithm I* using the BIDSAM algorithm for model selection has a lower value of the objective function given by Eq.(3.26) [129,137] of -2936 as against -2874 for *Algorithm II* model. Validation plots in Fig. 3.23 for MCBIDSAM models and their corresponding $RMSE$ values in Table 3.6 show that both algorithms have minimized deviations from the truth by reducing the largest $RMSE$ values from 0.17 *kmol/hr* and 0.39 *kmol/hr* to 0.10 *kmol/hr* and 0.07 *kmol/hr* ,respectively. One key advantage of *Algorithm II* over *Algorithm I* is that this accuracy is achieved by a model with a closed form unlike for the case of *Algorithm I* where data reconciliation is required for its implementation. Note the results shown in this case study are for constraints formulated for satisfaction of mass conservation only at steady state conditions. Fig. 3.24 shows how *Algorithm II* which has a model with a closed form outperforms *Algorithm I* in minimizing the atom balance violations during transience since the latter swings to the enforcement of mass constraints only when input and output variables are considered to be steady. This difference has become more prominent in this case study as the input variables are not necessarily at fixed values due to control actions as shown in **Appendix B**.

(a)



(b)

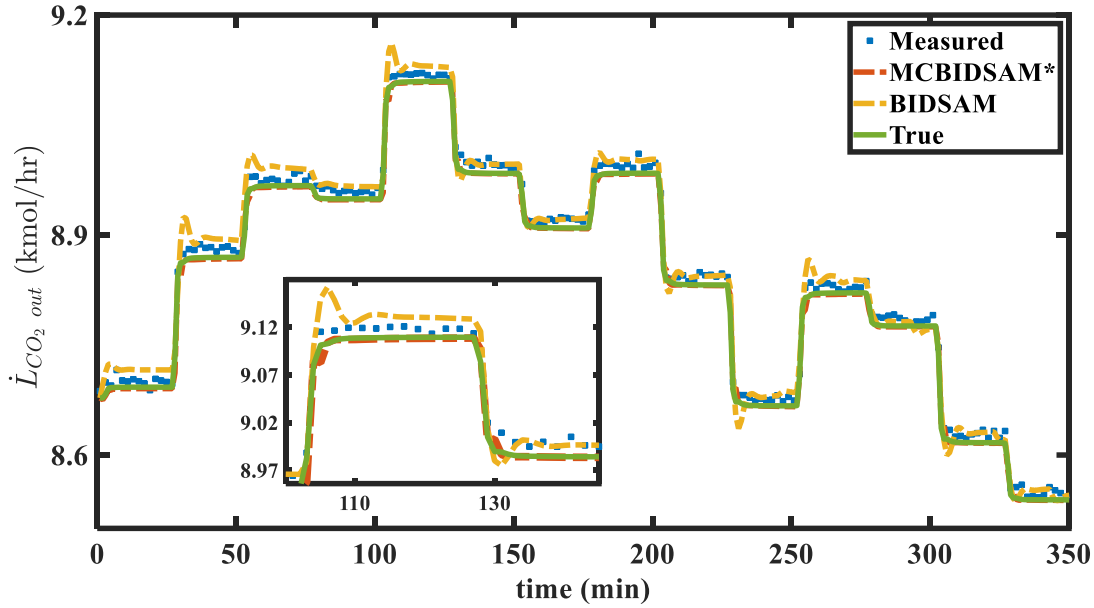


Fig. 3.23. Validation plots for the prediction of $L_{CO\ out}$ with models developed from noisy data with bias (a) Algorithm I (b) Algorithm II.

Table 3.6. Performance measures of the dynamic models of the absorber unit of CO_2 capture plant.

Algorithm	Model	No. of parameters	RMSE (*kmol/hr*)			
			$\dot{G}_{CO_2\ out}$	$\dot{G}_{H_2O\ out}$	$\dot{L}_{CO_2\ out}$	$\dot{L}_{H_2O\ out}$
Algorithm I	**Trained using *True* Data**					
	BIDSAM	36	7e-4	0.02	0.01	0.17
	MCBIDSAM*	40	7e-4	0.02	0.06	0.17
	Trained using Noisy Data with Bias					
	BIDSAM	28	1e-3	0.03	0.01	0.17
	MCBIDSAM*	32	1e-3	0.04	4e-3	0.10
Algorithm II	**Trained using *True* Data**					
	BIDSAM	48	5e-4	0.03	0.02	0.26
	MCBIDSAM*		9e-4	0.04	0.01	0.17
	Trained using Noisy Data with Bias					
	BIDSAM	44	6e-4	0.03	0.02	0.39
	MCBIDSAM*		8e-4	0.03	4e-3	0.07

Key:
MCBIDSAM*-Mass constrained without holdup information.

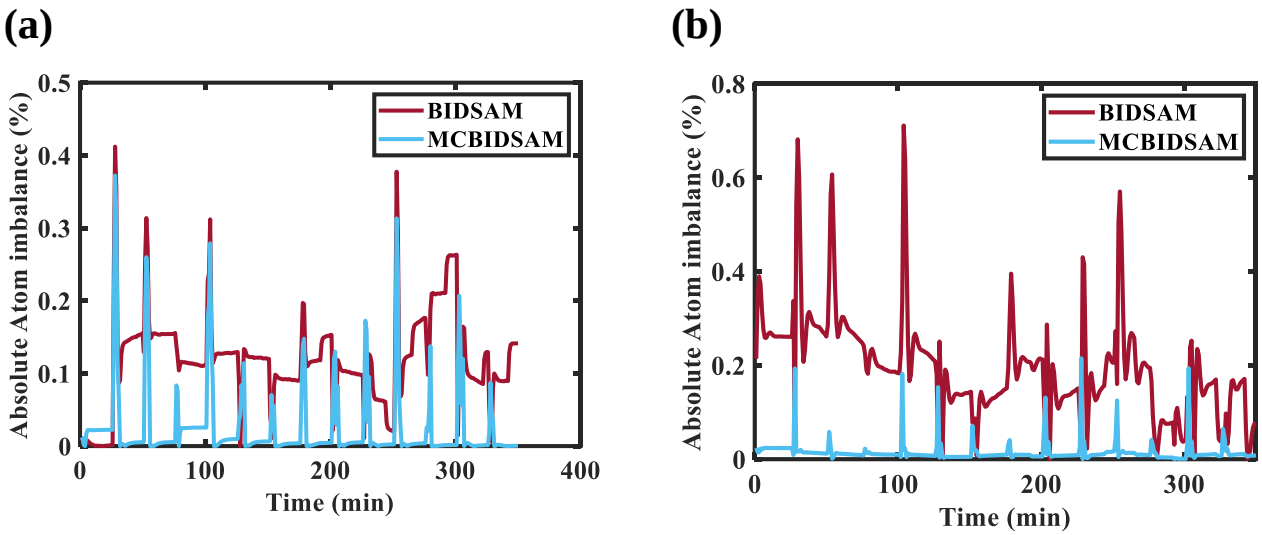


Fig. 3.24. Absolute percentage atom imbalance for validation of models for CO_2 absorber unit developed from noisy data with bias (a) Algorithm I (b) Algorithm II.

3.6 Conclusions

The proposed BIDSAM algorithm has been evaluated under different scenarios of noise and uncertainties. For the case of synthetic data, models are developed using noise-free data as well as data corrupted with cross-correlated and autocorrelated noise at different instances. The performances of the BIDSAM models are compared with state-of-the-art approaches which our proposed algorithm outperform in all instances in terms of model sparsity and robustness to noise. Real plant data with uncharacterized noise from a pilot post combustion CO_2 capture unit were also considered for development of fully data driven model and the BIDSAM algorithm demonstrated outstanding performance compared to the SINDy algorithm. Finally, the proposed BIDSAM algorithm was used to develop models used in hybridization with first-principles model of an industrial superheater of a coal-fired power plant. Again, the algorithm demonstrated remarkable robustness by yielding predictive models that not only account for uncertainties that are not captured in the first-principles model, but also facilitate spatial resolution through accurate predictions of steam temperatures in individual tubes along the superheater.

Two algorithms are developed for the enforcement of constraints to guarantee the satisfaction of mass conservation laws by sparse data driven models that approximate system nonlinearities with linear combination of basis functions. The BIDSAM algorithm developed in our previous work was used for the building of sparse models that are robust to the presence of correlated noise in the training data. While MCBIDSAM *Algorithm I* augments the BIDSAM model with a data reconciliation step, *Algorithm II* proposes a strategy for deriving a set of equality constraints on the model parameters that will guarantee the satisfaction of the conservation laws. Two case studies are considered for both steady state and dynamic model building. In both case studies, when models are trained using the *true* data, both *Algorithms I* and *II* results to satisfaction of mass balance without significant loss of prediction accuracy of the original model. For the steady state scenario in case study 1, *Algorithm I* require 33% lesser number of parameters for similar performance to *Algorithm II*. On the contrary, for steady state modeling for case study 2, both algorithms attain similar performance with exactly same number of parameters when *true* and noisy data were used for model development.

For the dynamic modeling, results from two case studies conducted here show that both algorithms can guarantee the exact satisfaction of mass conservation at both steady state and during transience. In the first case study where holdup information was not available, exact satisfaction of mass constraints was guaranteed only at steady state. In the second case study, the exact satisfaction of mass constraints at both steady state and dynamic conditions resulted in at least 100% increase in model size when there are changes in reactor volume. An alternative approach of utilizing available hold up information is proposed and upon implementation, the maximum instantaneous atom imbalance was reduced from 17.2 to 10.6%. In general, *Algorithm I* results in model with fewer parameters compared to *Algorithm II*. The case studies show that both algorithms can achieve satisfaction of mass conservation laws by utilizing only known boundary conditions for their implementation, thereby making them applicable to many systems. Another key advantage of the proposed MCBIDSAM algorithms is that they preserve sparsity and interpretability of the original models identified from the BIDSAM algorithm. Future work would be undertaken to ensure the simultaneous satisfaction of mass and energy conservation laws by dynamic sparse models both at steady state as well as during transient conditions.

4 Development and Validation of the Neural Network and Gaussian RBF Model

4.1 Introduction

The section aims at the development of hybrid first principles – artificial intelligence approaches for online health monitoring of complex nonlinear dynamic chemical processes. While first-principles models can provide very good predictions even for cases when there are no data at all, or data are limited in certain range of operating conditions, the development of accurate first-principles models for complex nonlinear dynamic systems can be time consuming, computationally expensive, and may be infeasible for certain systems due to lack of sufficient knowledge (information). It is also challenging to adapt first-principles models for time-varying systems. On the contrary, black-box or data-driven models are relatively easier to develop, simulate and adapt online[164]. Over the last few decades, numerous advantages of using data science and artificial intelligence (AI) / machine learning (ML) approaches in the field of chemical engineering have been exploited and analyzed. Neural networks (NNs) have been identified as one of the many different artificial intelligence approaches that exhibit a strong potential to modeling highly complex nonlinear dynamic systems[165].

In addition to the Bayesian machine learning approaches developed in Section 2, this section also focused on the development, training and applications of novel deterministic and stochastic dynamic physics-constrained hybrid neural network structures and hybrid first principles-NN models for data-driven modeling of complex nonlinear dynamic chemical process systems. The NN models developed and proposed as part of this research can be either implemented independently in a fully data-driven approach or be synergistically hybridized with other first-principles (physics-based) models for exploiting their key strengths. Most of the AI/ML techniques, especially neural networks, have evolved since the last few decades for a diverse range of chemical engineering applications such as data classification, fault detection and diagnosis, chemical process design, monitoring, and control. Although a plethora of different types of deep and shallow NN models have been successfully implemented for modeling various complex nonlinear transient chemical processes, the successful development of such techniques requires large data sets which may not often be available for modeling various chemical engineering systems. This work develops novel architectures and training algorithms for hybrid all-nonlinear series and parallel static-dynamic NN models for applications to chemical systems[166]. The sequential decomposition-based training algorithms thus formulated exploit the model structure, leading to independent and separate training of static and dynamic networks by different parameter estimation algorithms while still achieving optimal convergence by solving an outer layer optimization. The all-nonlinear series and parallel networks have been shown to significantly outperform typical deep recurrent neural networks, with the proposed sequential training algorithms leading to 50-100 times faster computation than simultaneous training algorithms.

One of the more recent advancements in the field of process modeling using NNs points to the augmentation of various physics conservation laws pertaining to a system while constructing optimal network models during both training and simulation. The primary motivation of such approaches stems from the fact that the measurement / experimental data available for training may not necessarily satisfy mass and energy balance equations and/or other thermodynamics / physics-based constraints. If such conservation laws are not considered during machine learning, model predictions can violate system physics and hence are not meaningful. However, in almost all existing literature on physics-informed neural networks[167] (PINNs), the physics conservation equations have been augmented in the loss

function as additional penalty terms, thus serving as soft constraints without ensuring an 'exact' satisfaction of the conservation laws. This work focuses on the development of algorithms for exactly satisfying physics conservation laws such as mass and energy balances as well as thermodynamics constraints during both training and forward problems using noisy transient data. The mass-energy-thermodynamics constrained neural network (METCNN) models developed in this research are found to be very accurate capturing the system truth with minimum bias for all examples that are evaluated, even when the training data are corrupted with uncertainties[168–170].

For modeling uncertain systems, probabilistic neural networks can be highly useful. However, synthesis of optimal hybrid networks and parameter estimation for probabilistic neural network models of dynamic uncertain systems without suffering from the 'curse of dimensionality' is considerably challenging. It is also challenging to satisfy physics constraints when probabilistic NNs are used. The final set of model architectures and algorithms developed in this research are aimed at optimal integration of stochastic and conventional NNs as well as efficient formulation of training algorithms for hybrid series and parallel constrained stochastic-deterministic network models. The corresponding approaches proposed in this work for construction of optimal parsimonious hybrid stochastic networks exhibit considerably superior results compared to the state-of-the-art approaches when evaluated for large-scale complex nonlinear process systems. All proposed data-driven / hybrid models and algorithms are evaluated and validated for modeling various complex nonlinear transient noisy chemical process systems. It is worth mentioning that the proposed data-driven models can either be implemented independently by themselves or hybridized with appropriate first-principles models during implementations while modeling complex nonlinear transient chemical process systems.

4.2 Hybrid All-Nonlinear Series / Parallel Static-Dynamic Neural Networks

The use of NN models has gained significant attention in modeling and control of complex nonlinear chemical processes. Traditional NN architectures, however, may struggle to capture the inherent dynamic and static nonlinearities present in various chemical systems. Motivated by the existing gaps in open literature with respect to the state-of-the-art models, this chapter discusses the development of hybrid NN architectures combining nonlinear static and nonlinear dynamic network models in series and parallel configurations[166]. Sequential parameter estimation algorithms have been proposed which provide sufficient flexibilities of training the static and dynamic networks separately and independently, while solving an outer layer optimization to ensure overall convergence. The following sub-chapters extensively account for the limitations of existing NN models for nonlinear process systems, followed by the detailed descriptions of the hybrid models and training algorithms developed in this work, along with the corresponding results obtained for a number of different case study examples.

4.2.1 Literature Review

Various types of AI modeling techniques such as NNs, expert systems and fuzzy logic have been employed for process synthesis, design and modeling[164]. One of the earliest approaches that has been applied to various chemical engineering problems is the static NN that have found widespread applications in system identification and control[171], data classification[172], fault detection and diagnosis[173,174] and pattern recognition[175]. Static NNs are often represented by multilayered feed-forward neural networks with one or more hidden layer of neurons where parameters are typically estimated using the backpropagation algorithm[176–178]. However, the static ANN may face difficulty in accurately representing the dynamic systems[179]. Furthermore, the static networks can also be inefficient in

representing temporal systems having time-lagged input variables[180]. Highly complex nonlinear dynamics involved in many process systems have motivated the development of dynamic and adaptive ANN structures such as recurrent neural networks (RNNs). RNNs are structurally similar to conventional feedforward NNs, but also include weighted feedback (recurrent) connections[176] thus helping to model the transient response[181]. In this work, the RNN model is represented by the Nonlinear Auto-Regressive with eXogenous inputs (NARX)[182,183]. The NARX-type RNN is a powerful class of time delay RNN used for predicting arbitrary nonlinear time series mappings. A number of different applications of dynamic (recurrent) neural networks has been reported in the open literature, such as modeling and control of nonlinear reactor systems[184,185] and optimization of power systems[186], etc. RNNs can be modified to generate adaptive neural network structures like LSTM which enhances the performance of the RNNs by storing information for long or short durations of time and preventing the problems of vanishing gradients during parameter estimation[187]. While existing adaptive NN models such as adaptive bidirectional associative memory[178] and transversal / recursive filters[188,189] have found applications in signal processing and communication, they may be inadequate for many chemical engineering applications where learning must be accomplished with data that may be noisy, limited, and time-varying.

For many nonlinear dynamic systems, it can be difficult to adequately model them by a simple (i.e., containing just one or very few hidden layers) static or dynamic neural network[190]. If large number of hidden layers are used, it can lead to potential overfitting during model training especially in absence of large data sets[191,192]. This is an issue especially for many chemical engineering systems for which large training datasets including dynamic data of desired input/output variables are not publicly available. For such systems a hybrid static-dynamic network can be useful. Two hybrid block-oriented structures that have been widely used are the Hammerstein and Wiener models including their many variants such as Hammerstein-Wiener and Wiener-Hammerstein type of models. The Hammerstein type of structure consists of a nonlinear static model followed by a linear dynamic model, whereas the Wiener type is basically the reverse, i.e., a linear dynamic model followed by a nonlinear static model. The Hammerstein, Wiener, Hammerstein-Wiener, and Wiener-Hammerstein type of models have been widely used for representing many chemical process systems such as a continuous stirred tank reactor (CSTR) system[193], a pH neutralization process[194], a polymerization reactor[195], distillation towers[196,197], etc. In those works, static nonlinearities are represented by different types of nonlinear models such as polynomial functions[196], ANNs[198], wavelets[197], etc. while the linear time invariant (LTI) dynamic model is typically represented by state space model[198,199], transfer functions[200,201], Laguerre models[202], finite impulse response models[203], etc. An excellent example of the Wiener structure used for several chemical engineering applications where the nonlinear model is given by the static single hidden layer perceptron and the linear dynamic model is given by a set of discrete Laguerre systems is DABNet (Decoupled A-B Network)[204,205]. DABNet has been successfully applied to many applications including the identification of a two-tank pH neutralization reactor, a continuous polymerization reactor process, and a solid sorbent-based CO_2 capture process[206]. Furthermore, the proposed network structures are compared with a couple of other widely used data-driven models for system identification, the Long Short Term Memory[207] (LSTM) type and the Gated Recurrent Unit (GRU) type of RNNs[208,209]. For the case studies considered in this work, performances of the proposed hybrid series and parallel network models have been compared with LSTMs, GRUs, and DABNet, in addition to the comparison with respect to the static-only and dynamic-only network architectures.

Existing literature lacks work on synthesizing network structures where both static and dynamic networks are nonlinear[210]. While there are several examples in the existing literature where a linear dynamic model integrated with a nonlinear static neural network has been used in Hammerstein-type or Wiener-type structures[198,204,205,210,211], these models may suffer from the curse of dimensionality or excessive computational expense during parameter estimation, especially while modeling systems with highly nonlinear process dynamics[212]. For data-driven models of dynamic systems, one key measure of performance is instantaneous error[213], which does not get appropriately captured by mean squared error (MSE) or other similar measures. High instantaneous errors are unacceptable for many process applications such as process monitoring, control, dynamic optimization, etc. that would utilize instantaneous prediction from the data-driven model to take important actions and make critical decisions. Therefore, high value of instantaneous error can eventually lead to many undesired conditions such as unsafe operation, product quality violation, unacceptable performance loss, etc. For a measure of instantaneous error, we consider maximum absolute instantaneous error (MAIE). Moreover, in addition to the typical objective function aimed at minimizing the mean squared error (MSE) during model training, a second objective is considered in this work which penalizes overfitting and parametric uncertainties by minimizing the corrected Akaike Information Criteria (AICc) along with a covariance complexity penalty term[214]. To reduce errors in data-driven models for highly nonlinear dynamic systems, we propose and investigate two fully nonlinear static and dynamic networks that are arranged in series. In one structure, the nonlinear static network is followed by a nonlinear dynamic network (abbreviated as NLS-NLD). In the other structure, the nonlinear dynamic network is followed by the nonlinear static network (abbreviated as NLD-NLS). In addition to the series networks, a parallel network is also investigated where nonlinear static and dynamic networks are arranged in parallel - abbreviated as the NLS||NLD. The review paper by Schoukens and Tiels[210] has noted that the parallel architecture with nonlinear dynamic models/networks is one of the unexplored area of research. With these motivations, architectures with nonlinear static and nonlinear dynamic NNs in parallel are also investigated in this work for modeling complex dynamic chemical engineering processes.

Training the hybrid all-nonlinear static-dynamic networks can be difficult. Complex deep neural networks using LSTMs, GRUs, etc. may need a large number of parameters and complex structures. Therefore, the primary motivation of this work is to develop relatively simpler hybrid network structures with significantly lesser number of parameters that can still adequately model highly nonlinear dynamic systems. Neural network training algorithms can be largely divided into two categories - first order methods such as variants of gradient descent, conjugate gradient algorithms and second order methods such as variants of Newton and quasi-Newton algorithms[215,216]. First order methods are extensively used for training large networks involving thousands of parameters but may require significant tuning of hyper parameters and can suffer from slow convergence. While second order methods can address some of the issues mentioned above, they can be computationally expensive due to Hessian calculation especially if either the static or the dynamic network or both are large. Irrespective of which method is used, training the dynamic network requires dynamic optimization that can be challenging when the network is large and large amount of dynamic data are considered for training[215,217,218]. Many efficient algorithms have been developed that exploit the specific structure/type of the static and dynamic network and fundamental differences between steady-state and dynamic optimization in general. Even when the similar algorithm for training the static nonlinear network is used for training the dynamic network, there are usually considerable modifications for training the dynamic network. For example, classical gradient descent algorithms using backpropagation algorithm can be applied for training nonlinear static and dynamic networks, but considerable modifications have to be made for training the dynamic network[219]. While the parameter estimation for both static and dynamic networks can be

undertaken simultaneously by discretizing the time domain, for example, and by using a single training algorithm such as the Newton or quasi-Newton based method, such approaches can be inefficient and computationally expensive as the efficient algorithms for training static and dynamic networks can be different for many systems. Therefore, in this work, algorithms are developed for training the series and parallel hybrid nonlinear static-dynamic networks where the static and dynamic networks can be trained separately by the same or different training algorithms. These algorithms are denoted as 'sequential' algorithms as the static and dynamic networks are trained in a sequence in contrast to the 'simultaneous' approach where the entire network comprising static and dynamic networks are trained simultaneously. Performances of the proposed structures and algorithms have been evaluated for different nonlinear case study examples, including the steam superheater systems at partner power plants[147], pH neutralization reactor[204], Van de Vusse reactor[220,221], and a pilot plant for post-combustion CO_2 capture using the monoethanolamine solvent[222].

4.2.2 Neural Network Structures

While there is plethora of ANN structures, only the ones used in this research are briefly discussed here. More details about various structures can be found in several references[176,215]. The multilayered feedforward neural network (MLFFNN) is considered as the nonlinear pure static network in this work, while the recurrent neural networks (RNN) with weighted feedback connections are considered as the pure dynamic neural network. The chosen network structures are very common. The chosen network structures are arbitrary as the developed algorithms can be applied irrespective of the choice of the network structures.

Nonlinear Static (NLS) Network

The nonlinear static (NLS) neural network considered in this work is represented by a typical three-layered pure static feedforward network with bias. Mathematically, the network architecture can be expressed as[176]:

$$\boldsymbol{y}_{i,NN} = \varphi_o[W_o\, \varphi_h(W_h \boldsymbol{u}_{i,NN} + \boldsymbol{b}_h) + \boldsymbol{b}_o] \tag{4.1}$$

where, $\boldsymbol{y}_{i,NN}$ is the output vector (matrix) for the i^{th} neuron, $\boldsymbol{u}_{i,NN}$ is the input vector (matrix) for the i^{th} neuron, W_o is the connection weight matrix between the hidden and output layers, W_h is the connection weight matrix between the input and hidden layers, $\boldsymbol{b}_h$ and $\boldsymbol{b}_o$ are the bias vectors for the hidden and output layers, respectively, and $\varphi_h(\cdot)$ and $\varphi_o(\cdot)$ are the activation functions of the nodes in the hidden and output layers, respectively. In this work, the standard logistic sigmoid function is used as the activation function for both the hidden and output layers, given as:

$$\varphi_h(\boldsymbol{u}) = \varphi_o(\boldsymbol{u}) = \frac{1}{1 + e^{-\boldsymbol{u}}} \tag{4.2}$$

Nonlinear Dynamic (NLD) Network

Dynamic or recurrent neural networks (DNN or RNN) can be used to represent temporal processing models and applications by incorporating time lags in the input variables. The network size of RNN is relatively compact as compared to multilayered feedforward NN for the same approximation accuracy of dynamic systems, because the feedback connections largely prevent the need of data windows of time-delayed inputs[215]. In this work, the NLD model is represented by the NARX-type RNN. The nonlinear autoregressive with exogenous inputs (NARX) type of dynamic neural networks represent a class of time-

delay recurrent networks[223] with feedback. The NARX-type of RNN utilizes the memory of past values of predicted outputs, along with external input variables, for nonlinear time-series prediction[224]. Throughout this research, it is assumed that the NARX-type RNN models have zero input delays and one output delay. Mathematically, the model is formulated as a discrete time input-output recursive equation as:

$$\boldsymbol{y}_{i,NN}(t+1) = \varphi_o\left[W_o\,\varphi_h\left(W_h\boldsymbol{u}_{i,NN}(t+1) + \boldsymbol{b}_h + W_{fb}\boldsymbol{y}_{i,NN}(t)\right) + \boldsymbol{b}_o\right] \tag{4.3}$$

where, $\boldsymbol{y}_{i,NN}(t+1)$ refer to the predicted output from the NARX-type RNN at time $(t+1)$, W_{fb} refers to the connection weights corresponding to the feedback of the model outputs from previous time step, and all other variables / functions have the exact same interpretation as compared to Eq. (4.1).The activation functions, i.e., $\varphi(\cdot)$ have been taken as the logistic sigmoid function, similar to the previous sections. The NARX-type of recurrent neural networks provide much more flexibility as compared to the classical Hopfield network with respect to prediction accuracy, due to the presence of exogenous input data. The NARX networks with a finite number of parameters have constructively been proved to be at least as computationally efficient as the Turing machines[225].

Hybrid Series and Parallel All-Nonlinear Networks

Fig. 4.1 (a), (b), and (c) show configurations of NLS-NLD, NLD-NLS, and NLS||NLD, respectively, where $\boldsymbol{h}_i(t)$ is the vector of intermediate outputs in series network, and $\boldsymbol{y}_{s_i}(t)$ and $\boldsymbol{y}_{d_i}(t)$ are, respectively, the outputs from the static and dynamic models in NLS||NLD. In the hybrid parallel structure, i.e., NLS||NLD model, the same input vector/matrix, $\boldsymbol{u}_{i,NN}(t)$ is fed to each of the static and dynamic networks, while the respective outputs $\boldsymbol{y}_{\boldsymbol{s_i}}(t)$ and $\boldsymbol{y}_{\boldsymbol{d_i}}(t)$ are linearly combined to generate the overall network output $\boldsymbol{y}_{i,NN}(t)$, as:

$$\boldsymbol{y}_{i,NN}(t) = \boldsymbol{y}_{\boldsymbol{s_i}}(t) + \boldsymbol{y}_{\boldsymbol{d_i}}(t) \tag{4.4}$$

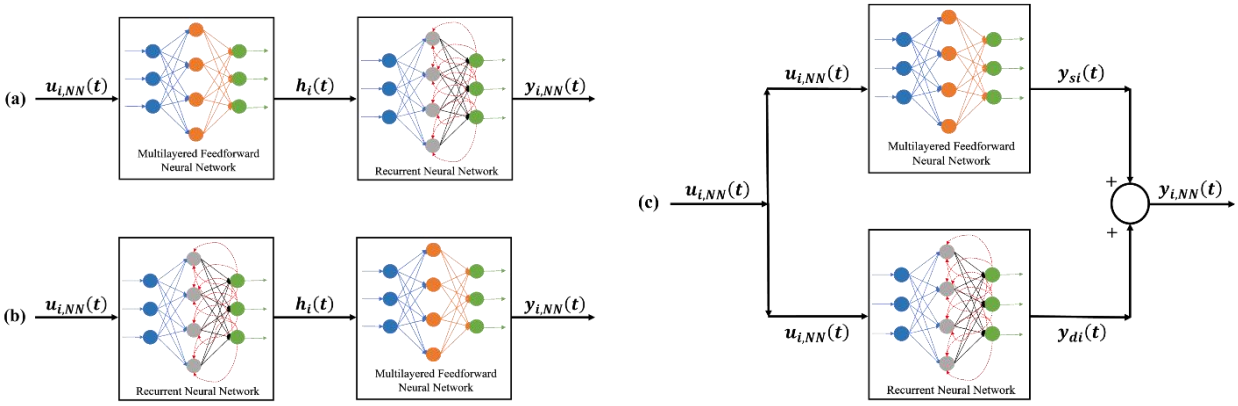


Fig. 4.1. Architectures of fully nonlinear networks: (a) NLS-NLD, (b) NLD-NLS, and (c) NLS||NLD

4.2.3 Training Algorithms for Hybrid All-Nonlinear Networks

For simplicity, algorithm development is described below for the common training objective that minimizes some form of squared error for both series and parallel hybrid structures. However, the algorithms still hold for other objective functions considered in this work.

Series Networks

The following proposition is the key to the proposed algorithm to solving both types of series networks. As an example, the proposition refers to NLD-NLS but can be applied to NLS-NLD as well. For the series network shown in **Fig. 4.2** (a), the training problem for the simultaneous approach is given by:

$$\min_{\theta_{stat},\theta_{dyn}} J = \left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_i\right)^{\boldsymbol{T}} R^{-1}(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_i) \tag{4.5}$$

Consider the torn network structure shown in **Fig. 4.2** (b). The sequential approach is given by the following optimization problems:

$$\min_{\theta_{stat},W_{int}} J_1 = \left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_i\right)^{T} R^{-1}(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_i) \tag{4.6a}$$

$$\min_{\theta_{dyn}} J_2 = \left(\boldsymbol{x}_{tar,i}(t) - \boldsymbol{x}_i(t)\right)^{T} \left(\boldsymbol{x}_{tar,i}(t) - \boldsymbol{x}_i(t)\right) \tag{4.6b}$$

$$s.t.\, \boldsymbol{x}_i(t) = W_{int}\boldsymbol{x}_{int,i}(t) \tag{4.6b}$$

where, $\boldsymbol{y}_{tar,i}(t)$ and $\boldsymbol{x}_{tar,i}(t)$ represent the target outputs from the nonlinear static and nonlinear dynamic models respectively, and R denotes the covariance matrix. All parameters (weights and biases) of the static and dynamic NN models considered in the hybrid structure are collectively referred to as θ_{stat} and θ_{dyn} respectively.

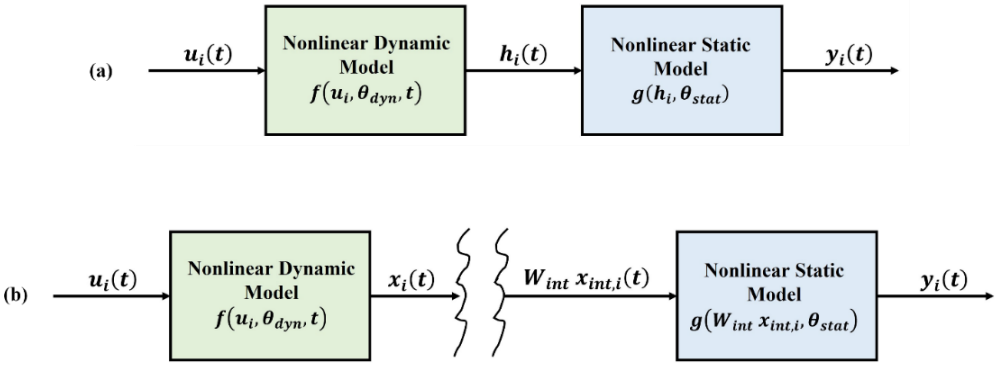


Fig. 4.2. Key to proving equivalence of optimization problems representing (a) simultaneous and (b) sequential approaches

For developing the algorithm for sequential optimization, we satisfy the equality constraint given in Eq. (4.6a)by the direct substitution approach. The proposed algorithm for training NLD-NLS is given in Fig. 4.3. The inputs to the nonlinear static network $\left(\boldsymbol{x}_{int,i}(t)\right)$ are randomly initialized at $k = 0$ whereas, the initial guess for the intermediate weights (W_{int}) is taken as a vector of ones. It is to be noted that the

algorithm for NLS-NLD is similar except for the sequence of solving the static and dynamic networks. For training NLD-NLS, the nonlinear static network is trained first followed by training of the nonlinear dynamic network. The equality constraint is exactly satisfied at each iteration by solving the linear constraint equations.

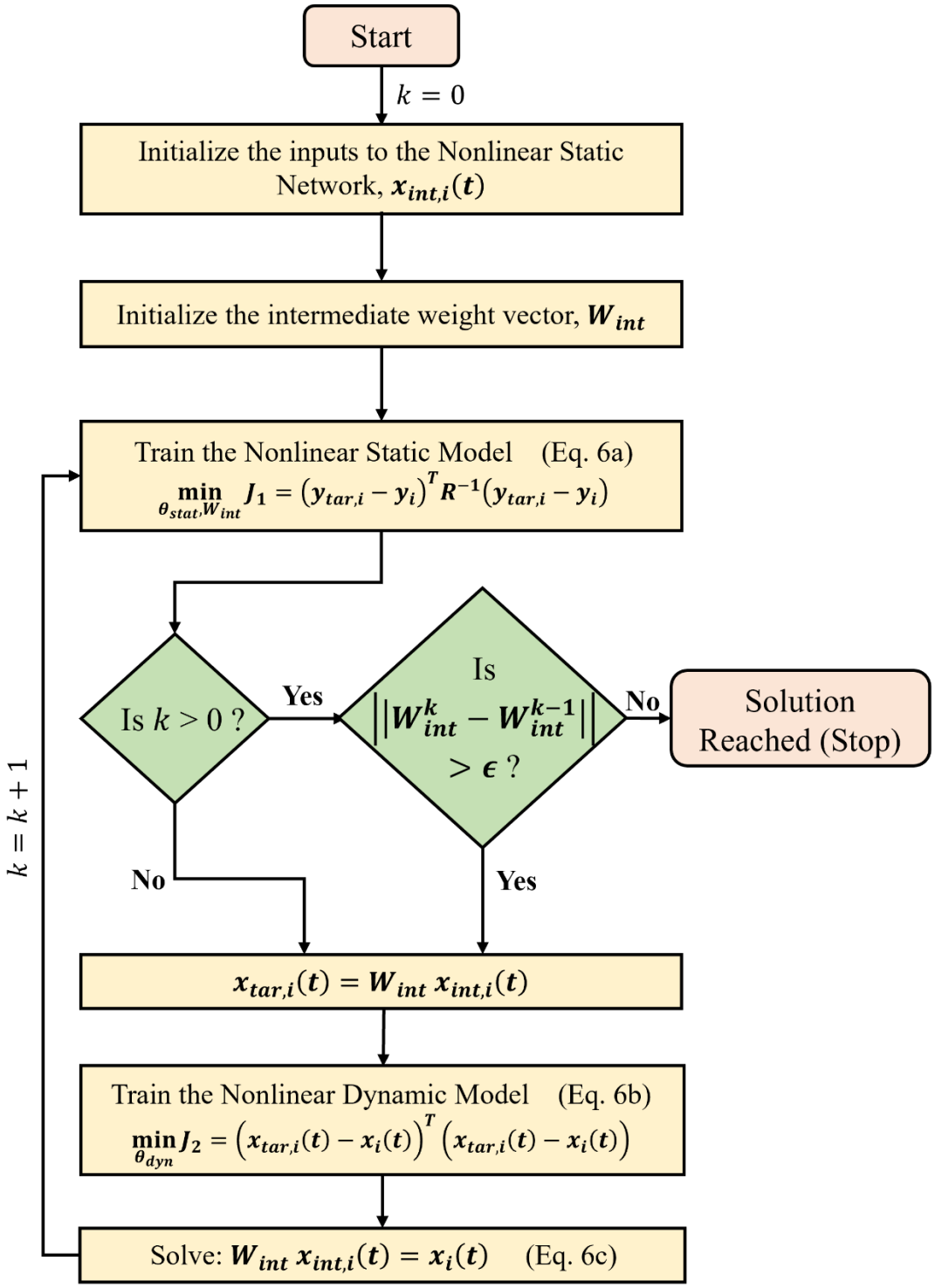


Fig. 4.3. Algorithm for Training Hybrid Series NLD – NLS Network Model

Parallel Network

The development of the parallel network (NLS||NLD) assumes that the outputs $\boldsymbol{y}_i(t)/\boldsymbol{y}_{i,NN}(t)$ are represented by outputs from separable and additive static and dynamic processes, as depicted in **Fig. 4.1** (c). For NLS||NLD, the input vector $\boldsymbol{u}_i(t)/\boldsymbol{u}_{i,NN}(t)$ is same for both the static and dynamic network. For the simultaneous optimization approach, the training problem for the hybrid parallel network shown in **Fig. 4.1** (c) is given by:

$$\min_{\theta_{stat},\theta_{dyn}} J = \left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_i\right)^T\left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_i\right) = \left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_{s_i} - \boldsymbol{y}_{d_i}\right)^T\left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_{s_i} - \boldsymbol{y}_{d_i}\right) \tag{4.7}$$

For developing the sequential approach, we decompose the network target outputs as:

$$\boldsymbol{y}_{tar,i}(t) = \boldsymbol{y}_{s_i,tar}(t) + \boldsymbol{y}_{d_i,tar}(t) + \varepsilon \tag{4.8}$$

where, $\boldsymbol{y}_{s_i,tar}(t)$ and $\boldsymbol{y}_{d_i,tar}(t)$ denote the target outputs of the static and dynamic networks, respectively. Then the sequential optimization problem is given by:

$$\min_{\boldsymbol{y}_{s_i,tar},\boldsymbol{y}_{d_i,tar}} J_1 = \left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_{s_i,tar} - \boldsymbol{y}_{d_i,tar}\right)^T\left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_{s_i,tar} - \boldsymbol{y}_{d_i,tar}\right) \tag{4.9a}$$

$$\min_{\theta_{stat}} J_2 = \left(\boldsymbol{y}_{s_i,tar} - \boldsymbol{y}_{s_i}\right)^T\left(\boldsymbol{y}_{s_i,tar} - \boldsymbol{y}_{s_i}\right) \tag{4.9b}$$

$$\min_{\theta_{dyn}} J_3 = \left(\boldsymbol{y}_{d_i,tar} - \boldsymbol{y}_{d_i}\right)^T\left(\boldsymbol{y}_{d_i,tar} - \boldsymbol{y}_{d_i}\right) \tag{4.9c}$$

For developing the sequential optimization algorithm, we set:

$$J_1 = 0 \;\; => \;\; \boldsymbol{y}_{d_i,tar}(t) = \boldsymbol{y}_{tar,i}(t) - \boldsymbol{y}_{s_i,tar}(t) \tag{4.10}$$

Therefore, the optimization problem given by Eq. (4.9a)can be written as:

$$\min_{\theta_{dyn}} J_3 = \left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_{s_i,tar} - \boldsymbol{y}_{d_i}\right)^T\left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_{s_i,tar} - \boldsymbol{y}_{d_i}\right) \tag{4.11}$$

Furthermore, we propose the following update scheme for iteration k:

$$\begin{aligned} \boldsymbol{y}_{s_i,tar_k} &= \boldsymbol{y}_{tar,i} - \boldsymbol{y}_{d_i,tar_{k-1}} = \boldsymbol{y}_{s_i,tar_{k-1}} + \boldsymbol{y}_{tar,i} - \left(\boldsymbol{y}_{s_i,tar_{k-1}} + \boldsymbol{y}_{d_i,tar_{k-1}}\right) \\ &\approx \boldsymbol{y}_{s_i,tar_{k-1}} + \boldsymbol{y}_{tar,i} - \boldsymbol{y}_{i_{k-1}} = \boldsymbol{y}_{s_i,tar_{k-1}} + \left(\boldsymbol{y}_{tar,i} - \boldsymbol{y}_{i_{k-1}}\right) = \boldsymbol{y}_{s_i,tar_{k-1}} + \boldsymbol{E}_{k-1} \end{aligned} \tag{4.12}$$

This leads to the algorithm given in Fig. 4.4.

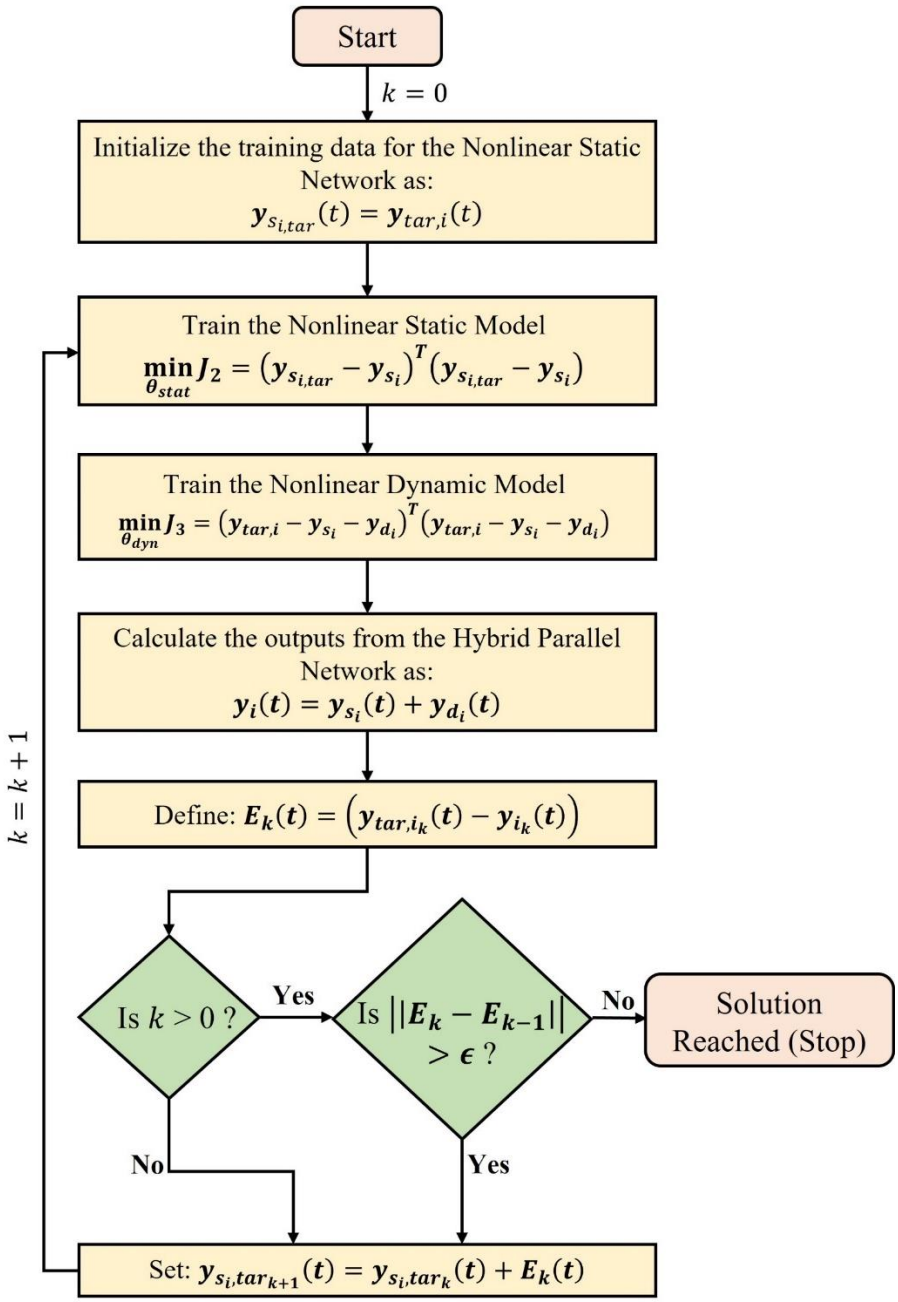


Fig. 4.4. Algorithm for Training the Hybrid Parallel NLS||NLD Network Model

Simultaneous Approach and Training of Static and Dynamic Networks

One straightforward approach to training the hybrid static-dynamic neural network architectures is to follow a 'simultaneous' approach, where all the weights are estimated simultaneously considering the entire system. The simultaneous approach leads to a nonlinear programming problem as the dynamic network models are discrete time. The fully Newtonian method with line search algorithm[226] is used for the simultaneous parameter estimation of the hybrid static-dynamic series and parallel networks. It was observed during training that numerical derivatives led to convergence issues for the simultaneous approach so analytical derivatives are used. The Levenberg-Marquardt approach is used for ensuring the positive definiteness of the Hessian. Various algorithms are evaluated for training the static and dynamic networks by themselves. The training algorithms for both the static and dynamic networks include the gradient descent (GD), scaled conjugate gradient (SCG), quasi-Newton method with line search and BFGS update as well as Levenberg-Marquardt (LM) algorithms.

Parameter estimation algorithms discussed above are available in many commercial optimization packages like the MATLAB Neural Network Toolbox, TensorFlow and PyTorch. However, we faced some difficulties in setting up those packages for implementing the flowcharts for series and parallel network training algorithms given in Fig. 4.3 and Fig. 4.4, respectively. The primary challenge faced while trying to implement the dynamic networks, specifically the modified Hopfield network model, in MATLAB Neural Network Toolbox, was the difficulty in modifying the MATLAB source code for including the exogenous inputs that we have incorporated in the hybrid model structure. It should be noted that the outer loop optimization problem in the proposed algorithm has to be coded by the user, even if some of the individual (i.e., static or dynamic) neural network models are trained by using an available toolbox, if the toolbox is flexible for desired modifications. Therefore, in this work, the MATLAB Neural Network Toolbox is used for training the nonlinear static network only, while the dynamic network is trained by using a user-developed a code in MATLAB that considers all modifications in the network structures proposed in the preceding sections. The user-developed code also makes it amenable to execute the proposed algorithms described above for sequential estimation. Performances of the proposed approaches in terms of computational expense and accuracy vary depending on the specific case studies and are provided below.

Training Objectives

Two different training objectives are considered in this work. The first one of them ('Objective 1') aims at minimizing the mean squared error (MSE) between the training data and neural network outputs for all the variables at all time steps, given as:

$$MSE = \frac{1}{n} * \frac{1}{t_n} \; * \; \sum_{i=1}^{n} \sum_{t=1}^{t_n} \left(\frac{\boldsymbol{y}_{tar,i}(t) - \boldsymbol{y}_i(t)}{\boldsymbol{y}_{tar,i}(t)} \right)^2 \tag{4.13}$$

where, n is the total number of output variables, t_n is the total number of time steps considered, $\boldsymbol{y}_{tar,i}(t)$ signifies the target or desired training data while $\boldsymbol{y}_i(t)$ represents the neural network output values. If the network size is large and/or sample size is small, 'Objective 1' can lead to overfitting and high uncertainty in parameter estimates. Therefore, another objective function ('Objective 2') given by Eq.(4.14) is also considered in this work. This objective considers the corrected Akaike Information Criteria (AIC_c) along with a covariance complexity penalty term. The AIC_c incorporates a bias correction term due to small

sample sizes assuming that the estimated parameters are normally distributed around the true parameter values[227]. Assuming that the parameters follow a multivariate normal distribution, the complexity penalty term can be written based on the work of van Emden[228] as:

$$AIC_c + 2 * C_{ve}(\sum) = -2\ln\left(L(\hat{\theta})\right) + 2k + \frac{2k\ (k+1)}{n-k-1} + \sum_i \ln(\boldsymbol{\sigma}_i^2) - \ln(\det\ (\sum)) \qquad (4.14)$$

where, $\left(L(\hat{\theta})\right)$ is the log-likelihood estimate evaluated at the maximum likelihood parameter values, $(\hat{\theta})$, k is the number of parameters characteristic of each model, n is the total sample size, $\boldsymbol{\sigma}_i$ denotes the diagonal elements of the covariance matrix $(\sum)$, and $\det\ (\sum)$ represents the determinant of the covariance matrix $(\sum)$. The additional covariance complexity penalty term is a measure of the information loss due to the assumption that the parameter estimates are independent[214]. While considering Objective 2, the neural network models have been initialized with the minimum number of possible parameters based on the model architecture, and the number of model parameters are increased sequentially after each iteration during model training.

4.2.4 Case Study: Dynamic Modeling

The performance of the candidate models and parameter estimation approaches discussed above are evaluated by implementing them for various complex nonlinear chemical process systems, namely the widely used two-tank pH neutralization system, Van de Vusse reactor, the industrial steam superheater systems at industrial partner power plants, and a monoethanolamine-based post-combustion CO_2 capture pilot plant at the National Carbon Capture Center at Wilsonville, Alabama, USA[222,229–232] among other case study examples. For all these case studies, input and output variables are normalized using the min-max normalization and 70% of the total dynamic data is used to train the NNs while the remaining is used for validation. k-fold cross validation (with k = 8) is performed for all three case studies. The network parameters (weights and biases) are initialized using Latin Hypercube Sampling (LHS). For a fair comparison, the same initial guesses for each static and dynamic neural network have been used to compare different candidate network models for a particular case study. In addition to MSE, MAIE is also calculated to compare the prediction performances of the different candidate neural network models for data at any given instance. MAIE (%) for the i^{th} output variable is defined as follows:

$$AMAIE_i = max\left(\left|\frac{\boldsymbol{y}_{tar,i}(t) - \boldsymbol{y}_i(t)}{\boldsymbol{y}_{tar,i}(t)}\right|\right) * 100\ \% \quad \forall\ t\ \in\ \{1, 2, \ldots\ldots, t_v\} \qquad (4.15)$$

where, t_v denotes the total number of time steps considered during validation. It was observed for all case studies that for NLS-NLD, NLD-NLS, and NLS||NLD, simultaneous approach and the developed sequential approaches lead to practically same values of parameters within the numerical tolerances for each specific type of hybrid model, thus leading to practically the same results for the validation data. For all case studies considered, NNs with a single hidden layer for both static and dynamic models have been considered. The number of neurons / nodes in the input and output layers of each NLS and NLD network considered in this work is exactly equal to the corresponding number of model input and output variables respectively, while the number of neurons in the hidden layer has been mostly considered in between half and twice the number of model inputs. Obviously, such network hyperparameters can be further optimized using outer layer optimization during model training. Furthermore, it was also observed for all case studies undertaken in this work that the sequential approaches incur much lower computational expense than their

simultaneous counterparts. This, in turn, indicates that the proposed sequential approaches would be more adequate if these black-box models were updated on-line. The number of neurons in the input and output layers are exactly equal to the total number of inputs and outputs for each specific system, except for some modifications in the dimension of the output layer for series networks based on the number of intermediate variables. The extensive discussion of results for all three case studies mentioned above can be found in our recent publication[166]. Among all, only the results obtained while modeling the post-combustion CO_2 capture pilot plant have been included here for brevity.

The dynamic modeling of the NCCC pilot plant[222,229–232] by data-driven approaches is a considerably challenging problem due to the highly complex dynamics of this electrolyte-based CO_2 capture process that include complex reaction kinetics, highly nonlinear vapor-liquid equilibrium, and complicated mass transfer mechanisms due to the presence of charged species. The absorber, given in Fig. 4.5, is decoupled, and modeled separately in this work. For this system, the neural network model input variables are taken as the flue gas (FG) inlet flow rate, flue gas inlet temperature, flue gas inlet CO_2 concentration (%v dry), lean solvent inlet flowrate and the lean solvent inlet temperature. Output variables are the clean (outlet) flue gas flow rate, clean flue gas outlet temperature and the clean flue gas CO_2 concentration (%v dry), thus generating a five-input-three-output neural network model. Here the experimental data from the pilot plant are used.

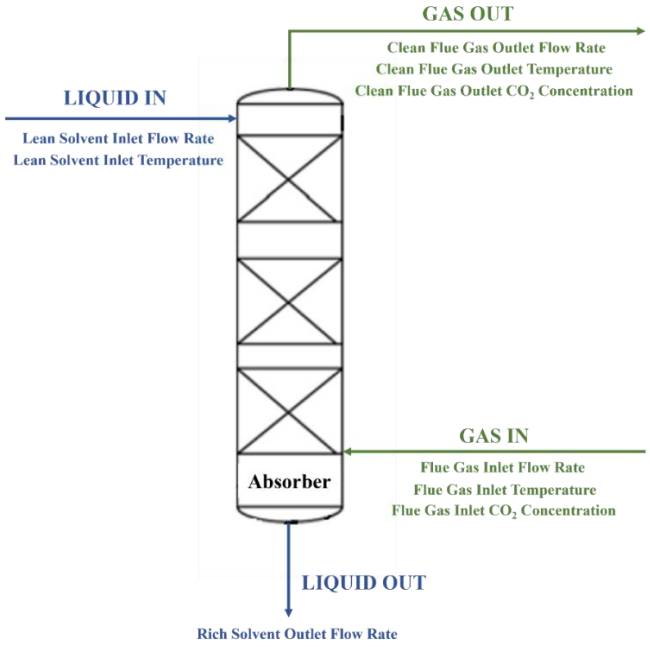


Fig. 4.5. Schematic of the absorber decoupled from the CO_2 capture pilot plant flow diagram

A summary of the results can be found in Table 4.1. The comparative analysis among the computational times recorded for training NLS-NLD, NLD-NLS, and NLS||NLD for this case can be found in **Table C. 1** in **Appendix C**. Detailed comparison and evaluation of training and validation MSEs using the two different objective functions have been included in Table 4.1.

Table 4.1: Comparison among the training algorithms, computational time and percentage MSE for both training and validation data of various types of models for the dynamic CO_2 capture pilot plant system w.r.t Objective 1 and 2

Types of Networks	Algorithm	Overall Computational Time		Percentage MSE for Training Data		Percentage MSE for Validation Data	
		Training using Objective 1	Training using Objective 2	Calculated Using Parameters Obtained by Minimizing Objective 1	Calculated Using Parameters Obtained by Minimizing Objective 2	Calculated Using Parameters obtained by Minimizing Objective 1	Calculated Using Parameters obtained by Minimizing Objective 2
Nonlinear Static Network	SCG	6.5 sec	61.2 min	0.08 %	0.1 %	37.1 %	21.2 %
Nonlinear Dynamic Network	LM	9.2 sec	47.5 min	0.05 %	0.08 %	10.1 %	7.7 %
DABNet	GD-GD (default)	20.1 min (1 pole), 42.9 min (2 poles)	--	0.05 %	--	27.3 % (1 pole), 22.6 % (2 poles)	--
LSTM1	Adam SGD (default)	17.5 min	--	0.15 %	--	6.4 %	--
GRU1	Adam SGD (default)	10.1 min	--	0.13 %	--	4.5 %	--
LSTM2	Adam SGD (default)	44.5 min	--	0.04 %	--	3.9 %	--
GRU2	Adam SGD (default)	45.0 min	--	0.03 %	--	3.1 %	--
NLS-NLD (this work)	BFGS-BFGS (sequential – proposed in this work)	48.7 sec	154.3 min	0.03 %	0.05 %	0.5 %	0.4 %
	Fully Newton (simultaneous)	51.8 min	--				
NLD-NLS (this work)	LM-SCG (sequential – proposed in this work)	64.4 sec	160.5 min	0.02 %	0.08 %	1.7 %	1.0 %
	Fully Newton (simultaneous)	59.2 min	--				
NLS||NLD (this work)	BFGS-LM (sequential – proposed in this work)	86.5 sec	120.8 min	0.05 %	0.09 %	0.9 %	0.6 %
	Fully Newton (simultaneous)	41.5 min	--				

The NLS-NLD model resulted in the best performance for this system, closely followed by NLS||NLD. Similar to the previous two cases, the difference in the computational time for training NLS-NLD, NLD-NLS, and NLS||NLD using the simultaneous approach and sequential approach is considerable as observed in Table 4.1. Table 4.1 also shows that the nonlinear dynamic network performs better than the DABNet, even when both the fast and slow poles are optimized for DABNet, probably due to the presence of high nonlinearities in measurement data from the carbon capture pilot plant. Moreover, NLS-NLD, NLD-NLS, and NLS||NLD yield an order of magnitude lower percentage MSE (0.5 %, 1.7 % and 0.9 %,

respectively) than DABNet (22.6 %) as well as require considerably shorter training time (49, 64, 87 sec, respectively), as compared to DABNet (20 and 43 min corresponding to one and two poles, respectively), LSTM1 (17.5 min), GRU1 (10.1 min), LSTM2 (44.5 min) and GRU2 (45 min). Furthermore, in comparison to both types of LSTMs and GRUs considered in this work, all series and parallel hybrid models show significant superior performance both in terms of percentage MSE / MAIE for validation data as well as overall computational time for training.

Fig. 4.6 compares the performance of NLS-NLD, DABNet, LSTM2 and GRU2 networks for the clean flue gas outlet flowrate with respect to Objective 1. The dynamics and nonlinearity in this variable certainly make it challenging to model this process accurately by a data-driven model. The NLS-NLD model still accurately captures the transient profile while large discrepancies are observed for the state-of-the-art models. Validation plots for other variables such as the flue gas outlet temperature and CO_2 concentration in the outlet flue gas stream are included in **Appendix C** (refer **Fig. C. 1** and **Fig. C. 2**). In order to achieve similar training performance compared to the proposed network models, the total number of parameters in LSTM2 / GRU2 has to be increased to around twenty times the number associated with LSTM1 / GRU1. However, for this case, NLS-NLD and NLS||NLD show a slightly superior performance as compared to NLD-NLS. The NLS-NLD model also outperforms DABNet significantly in terms of MAIE as seen in Table 4.2, while predicting all three dynamic output variables involved in this example. For all networks, it is further observed that the percentage MSE for the training data computed using the parameters optimized with respect to Objective 1 is lower than that computed using parameters obtained by minimizing Objective 2, whereas the MSE (%) for validation data follow the reverse trend due to better predictive capabilities of the parameters optimized with respect to Objective 2.

Table 4.2: Comparison in MAIE between NLS-NLD (optimal for this case study) and other state-of-the-art models for all outputs variables w.r.t. the dynamic CO_2 capture pilot plant system

Types of Networks	**Percentage MAIE for Validation Data**		
	Clean Flue Gas Flow Rate	**Clean Flue Gas Outlet Temperature**	**Clean Flue Gas CO_2 Concentration**
NLS-NLD (this work)	1.2 %	0.8 %	1.9 %
DABNet	5.7 % (1 pole), 4.1 % (2 poles)	9.8 % (1 pole), 7.6 % (2 poles)	71.5 % (1 pole), 66.7 % (2 poles)
LSTM1	1.6 %	3.5 %	16.2 %
GRU1	1.9 %	3.8 %	17.6 %
LSTM2	1.5 %	5.9 %	24.92 %
GRU2	1.8 %	2.0 %	8.72 %

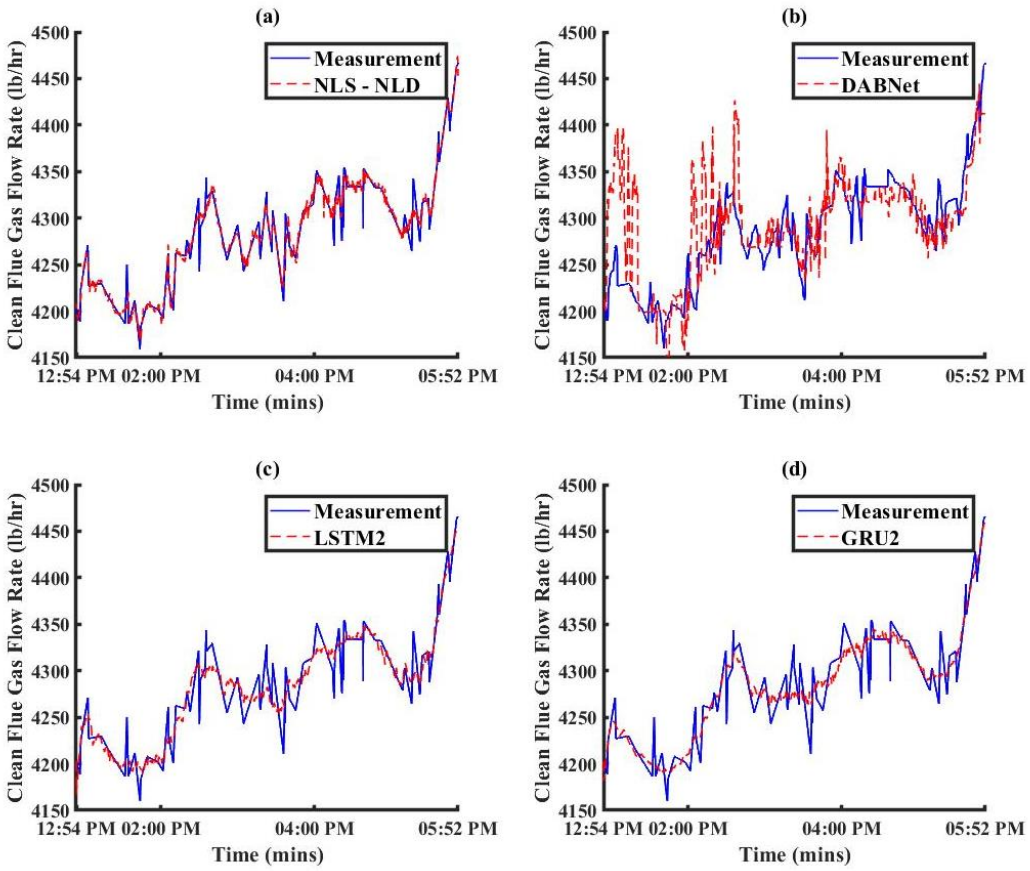


Fig. 4.6. Comparison between model prediction for validation data for the tower outlet clean flue gas flowrate for (a) NLS-NLD, (b) DABNet, (c) LSTM2 and (d) GRU2 networks

Table 4.3: Comparison among the optimal hybrid model and existing state-of-the-art models in terms of MSE during training and corresponding AICc values for CO_2 capture pilot plant

Types of Networks	**AICc**	**Percentage MSE for Training Data (Objective 1)**	**Percentage MSE for Validation Data (Objective 1)**
NLS-NLD (optimal for this case)	1101	0.03 %	0.5 %
DABNet	1478	0.05 %	27.3 % (1 pole), 22.6 % (2 poles)
LSTM1	1822	0.15 %	6.4 %
GRU1	1795	0.13 %	4.5 %
LSTM2	4038	0.04 %	3.9 %
GRU2	3465	0.05 %	3.1 %

Furthermore, Table 4.3 shows the comparison between the NLS-NLD (optimal for this case) model and DABNet, LSTMs and GRUs based on the AICc values calculated with the optimal parameters obtained from Objective 1. Similar to the previous case, the high values of AICc for LSTM2 and GRU2 indicate potential overfitting. It is also to be noted that the hybrid series and parallel all-nonlinear models record superior predictive capabilities while consisting of only around 90-100 parameters (weights and biases) as compared to the LSTM2 / GRU2 models which comprise approximately 650-700 parameters.

4.2.5 Comparing All-Nonlinear NN and Bayesian Machine Learning

Different types of ML models may have contrasting advantages and disadvantages, which become critical during the optimal selection of a specific data-driven model for a particular application as well as estimation of parameters during model training. This project also facilitated comparing and contrasting two different types of data-driven modeling approaches[233], namely the series and parallel all-nonlinear static-dynamic neural networks (developed as part of Subtask 2.4) and models obtained from a Bayesian ML approach (developed as part of Subtask 2.3). Both types of AI modeling approaches considered in this work have shown to significantly outperform several state-of-the-art steady-state and dynamic data-driven modeling techniques. While the all-nonlinear static-dynamic neural networks, along with their respective sequential parameter estimation algorithms, successfully achieve the trade-off between prediction accuracies and computational expense, the Bayesian ML (BML) approaches[234] offer capabilities for handling noisy data and uncertainty quantification in addition to enhanced model interpretability. The performances of the proposed model structures and algorithms have been evaluated for various nonlinear dynamic example problems including the industrial steam superheater systems, a plug-flow reactor involving vapour-phase cracking of acetone during production of acetic anhydride, and a pilot-plant for post-combustion CO_2 capture using monoethanolamine solvent. Similar to the previous subsection, the results obtained for the post-combustion CO_2 capture system have only been considered for comparing the two ML models here for brevity. Additional details on each of the data-driven approaches considered in this work as well as the comparative analyses of predictive capabilities of those models can be found in our recent publications[166,233,234].

In addition to comparing the training/validation RMSE and MAIE values, the NN and BML models have also been contrasted and compared in terms of number of model parameters to determine relative sparsity of the models, as well as the overall computational expense involved during model training. For all case studies in this work, both types of data-driven models provide multi-step ahead predictions of time-series without considering feedback from intermediate measurements, while modeling the dynamic nonlinear chemical process systems. It is also to be noted that for all dynamic case studies considered in this work, considerable dynamics in the model input variables are present and therefore, the sensitivity of model accuracy to dynamics in inputs are captured within the validation results presented in the document. Fig. 4.7 compares the performance of the NN and BML models while predicting the dynamic outlet temperature profile of clean flue gas coming out from the absorber tower. The highly nonlinear transients in this variable certainly make it challenging to accurately model the process using data-driven approaches. Similar plots for the outlet flue gas flow rate and outlet CO_2 concentration have been included under **Fig. C. 3** in **Appendix C**. A summary of the results can be found in Table 4.4.

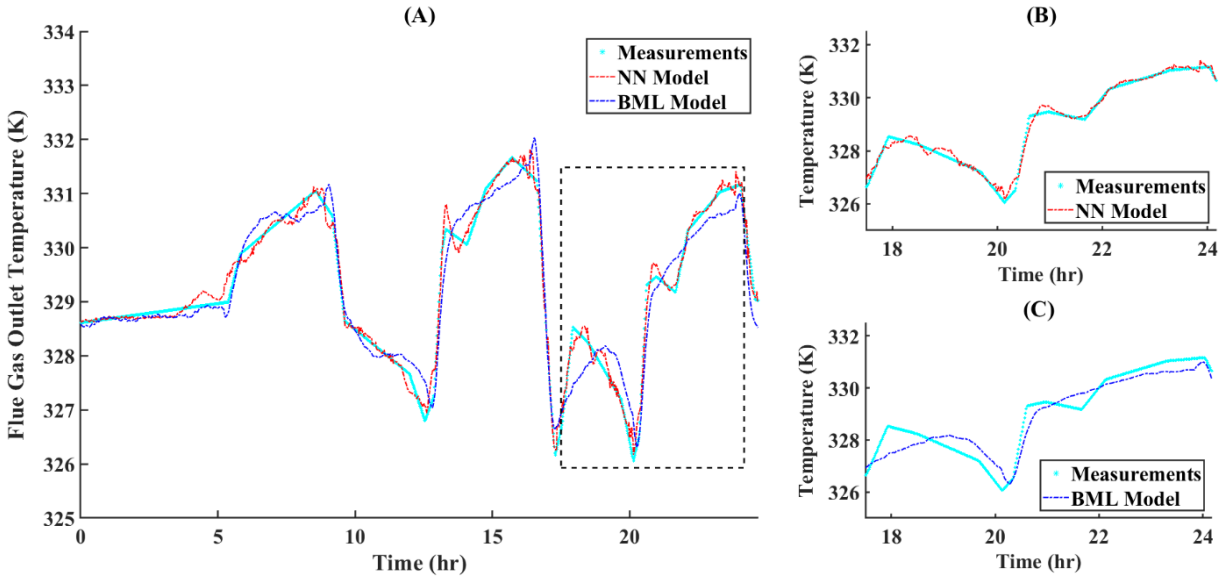


Fig. 4.7. Comparison between predictions obtained from the optimal NN and BML models for flue gas outlet temperature

Table 4.4: Comparison between NN and BML models in terms of overall RMSE / MAIE for the validation data, training computational time and number of parameters

Model Output Variables	NN Model		BML Model		Number of Model Parameters		Training Computational Time	
	RMSE	MAIE	RMSE	MAIE	NN	BML	NN	BML
Flue Gas Outlet Temperature	0.16 K (0.05%)	0.60 K (0.18%)	0.34 K (0.10%)	1.07 K (0.33%)	75	36	0.8 min	38 min
Flue Gas Outlet Flow Rate	21.36 $\frac{Kg}{hr}$ (1.07%)	119.88 $\frac{Kg}{hr}$ (5.62%)	33.65 $\frac{Kg}{hr}$ (1.75 %)	180.11 $\frac{Kg}{hr}$ (9.26%)				
Flue Gas Outlet CO_2 Concentration	0.05 %v (5.00%)	0.23 %v (29.57%)	0.13 %v (14.14%)	0.34 %v (47.43%)				

From Fig. 4.7 (A), it can be observed that both types of data-driven models capture the trends of the nonlinear transients in the flue gas outlet temperature profile to a satisfactory accuracy, yielding around 0.05% and 0.10% RMSE obtained by using optimal NN and BML models, respectively. However, the optimal NN model shows slightly better performance than the optimal BML model in accurately capturing the overshoots and undershoots in the transient temperature profile of flue gas, as can be seen in the magnified time-windows illustrated in Fig. 4.7 (B) and Fig. 4.7 (C). If the data-driven models developed in this work are used in a model-based control such as in nonlinear model predictive control, then an accurate prediction of flow rate, and composition is highly valuable for achieving desired capture target

in the CO_2 capture plant. As the outlet temperature can affect the solvent loss from the absorber, an accurate estimate of the outlet gas temperature can be useful for a predictive monitoring approach for the capture plants.

Furthermore, for all output variables, the NN model consistently records superior RMSE and MAIE values as compared to BML at a significantly faster computation rate during model training, as can be seen in Table 4.4, but at the cost of more than twice the number of parameters as considered by the BML model. This, in turn, leads to a slightly higher value of AICc (1207) for the optimal NN model obtained while modeling the absorber system under consideration when compared to the optimal BML model (1157), thus leading to superior model sparsity for BML. Overall, both NN and BML models yield less than 2% RMSE for prediction of the outlet flue gas temperature and flow rate, while the MAIE values obtained using the BML models for all output variables considered in this case study are nearly double of those obtained by using the NN models. In addition, the BML model obtained from the BIDSAM algorithm leads to enhanced model interpretability and extrapolative capabilities. Additionally, the optimal values of parameters for both NN and BML models for this case study have also been included in **Table C. 2** in **Appendix C.**

In summary, both types of data-driven models under consideration show satisfactory performance in modeling different complex nonlinear dynamic chemical process systems in terms of achieving the trade-off between predictive capabilities, model sparsity, and computational expense. For the post-combustion CO_2 capture pilot plant example, the optimal NN model recorded less than 5% RMSE for all the output variables at a faster computation rate which was significantly superior as compared to the optimal BML model but considered almost twice the number of parameters as that associated with BML. Furthermore, the overall computational expense associated with the training of the BML model has been shown to be consistently higher as compared to the NN model due to the associated MINLP approach within BIDSAM for selection of optimal set of candidate basis functions during parameter estimation. This computational time can however be further reduced by parallel computation of the BIDSAM algorithm while searching for the optimal subset of basis functions. For the hybrid all-nonlinear static-dynamic NN, the sequential training algorithms ensure faster convergence, even for highly nonlinear transient data. It should be noted that, the overall predictive accuracy of the data-driven models is critical for a variety of process applications such as model-based process controls, dynamic optimization, and process monitoring. One of the major limitations of the developed data-driven models may be attributed to the assumption that the model inputs are noise-free. Moreover, all possible types of uncertainty characterizations for the data have not been considered in this work. Future work would involve further algorithmic development to address additional noise characteristics in both input and output data, as well as the imposition of physics constraints during optimal training of the proposed AI models. Finally, the authors would like to point out that it is neither realistic nor desirable to identify superiority of a particular type of data-driven modeling approach over another since the efficiency of such black-box approaches largely depend on several factors including the specific system characteristics to be modeled, type and characteristics of data available for training, structures of the desired data-driven models, training algorithms and associated computational expense.

4.2.6 Conclusions

This chapter evaluates fully nonlinear hybrid static-dynamic NN models arranged in series and parallel for data-driven modeling of highly nonlinear dynamic systems. As opposed to training the networks simultaneously by a single optimization algorithm, algorithms are developed for sequential training of the networks enabling training of the static and dynamic networks by different optimization algorithms. It was observed that NLS-NLD, NLD-NLS, and NLS||NLD perform significantly superior compared to the static-only, dynamic-only networks as well as some of the most widely used state-of-the-art network models like DABNet, LSTMs, and GRUs, especially for systems with high nonlinearities in process dynamics like the CO_2 capture system. For all examples discussed in this work, the training of the hybrid series and parallel networks with respect to Objective 2 results in higher computational expense due to covariance calculations but results in better prediction accuracy. For the CO_2 capture pilot plant, there were considerable differences in training time. While NLS-NLD takes 48.7 sec for training, DABNet (2 poles), LSTM2 and GRU2 take 42.9 min, 44.5 min and 45 min, respectively for training yielding the similar MSE as NLS-NLD for the training data. Percentage MSE obtained using NLS-NLD for the validation data is 0.5% while that for DABNet (2 poles), LSTM2 and GRU2 are 22.6%, 3.9% and 3.1% respectively. Out of the output variables considered, clean flue gas CO_2 concentration was found to be the most challenging to predict. Percentage MAIE value for this variable obtained using for the validation data is 1.9% while that for DABNet (2 poles), LSTM2 and GRU2 are 66.7%, 24.92% and 8.72%, respectively.

It is also observed that the proposed algorithms for sequential optimization led to considerably faster convergence compared to the simultaneous approach with practically no difference in the optimal estimates of parameters. Moreover, the sequential approaches also facilitate the training of the different networks using different training algorithms, as well as assigning relatively superior initial guesses to the individual model structures. In addition to standalone data-driven modeling applications, the hybrid all-nonlinear static-dynamic NN models in integration with FP models exhibited enhanced model predictive and extrapolative capabilities of the overall hybrid FP+AI models, as compared to the standalone FP model for various nonlinear transient chemical process systems. While the optimal network and the optimal combination of optimization algorithms may greatly vary depending on the specific process to be modeled, this study provides a general framework that enables sequential solution of various combinations of structures using different training algorithms. Finally, by no means, it must be indicated that the comparisons presented in this work with respective to network structure and training algorithms are exhaustive since currently there are numerous structures and training algorithms in the existing literature. Future work may involve undertaking more such studies while comparing additional NN structures and algorithms.

4.2.7 Code Availability Statement

All data used in this work have been included in the form of plots. The MATLAB codes developed for training and validation of the proposed hybrid all-nonlinear series and parallel static-dynamic NN models can be publicly accessed on GitHub at: https://github.com/mukherjeeangan26/Hybrid_StaticDynamic_NN_Models

4.3 Mass-Energy-Thermodynamics Constrained Neural Networks (METCNN)

Although the hybrid all-nonlinear static-dynamic neural network models discussed in the previous chapter significantly outperformed several state-of-the-art models for all nonlinear dynamic chemical process systems considered, the corresponding training / validation data typically obtained from laboratories, pilot plants, or commercial plants may not necessarily satisfy mass and energy balances and other physics of the system. If physics constraints are not considered during optimal model synthesis, parameters can be biased. Even if the experimental data exactly satisfy physics constraints, the model may still not be able to satisfy physics constraints unless there is a 'perfect match' with the training data which is quite unlikely in most cases. Therefore, this project also focused on the development of physics-constrained neural network models aimed at exactly satisfying the mass and energy conservation of chemical process systems, as well as appropriate thermodynamics constraints. Algorithmic capabilities have been developed to ensure exact conservation of system physics during both the training and forward problems under various types of uncertainty characterizations during both steady-state and dynamic modeling implementations.

4.3.1 Literature Review

Over the last few decades, NNs have found numerous implementations in chemical engineering systems. They have been used for process design, monitoring, and control[164,171,235] with applications in varieties of systems such as computational fluid dynamics[236–238], reactive systems[166,204,233], distillation towers[196,197], thermofluidic processes[239–241]. In general, the data acquired from industrial plants, laboratories, and pilot plants do not necessarily satisfy the physics conservation laws, specifically the mass and energy balance equations as well as certain thermodynamics constraints. If the exact satisfaction of appropriate physics constraints is not ensured during the development of optimal data-driven models for chemical systems, the resulting predictions obtained may still not satisfy system physics during parameter estimation (interchangeably referred to as the 'training problem' in this report) and more significantly, during model simulation (interchangeably referred to as the 'forward problem' in this report). Thus, any decisions (e.g. design, control, monitoring, prediction, etc.) made by using such a model can suffer from unacceptable error and may not be physically realistic.

One of the approaches for incorporating physics-based knowledge during optimal syntheses of neural network models is denoted as physics-constrained[242] or physics-informed[167,243] neural networks (PCNN or PINN). Typical PINNs impose the physics conservation laws as soft constraints during formulation of the respective training problems through additional penalty terms in the objective function, thus ensuring 'approximate' satisfaction of the system physics. Furthermore, such models may also require rigorous understanding of the underlying chemical process(es), thus leading to highly system-specific implementations. However, it must be mentioned that for most (if not all) chemical engineering processes, it is highly desired that specific mathematical relationships, such as those represented by mass and energy conservation laws as well as system thermodynamics equations, are 'exactly' satisfied, instead of approximately[168,239]. Additionally, most of the papers in existing literature focused specifically on exact conservation of system physics[239,244,245] develop algorithms that are either system-specific or require significant physics-based (first-principles) information / knowledge about the respective process under consideration or can only accommodate linear constraint equations as a post-processing step. However, all physics constraints pertinent to a given system need not be necessarily linear. Moreover, it is also expected that the optimal physics-constrained data-driven models guarantee the 'exact' conservation of system physics both during the training problem, and more essentially, during the forward problem (model simulation), especially when the training data are corrupted with varying types of

uncertainties and / or biases[243]. In two of our recent publications, we have demonstrated how the exact satisfaction of mass and energy balance constraints can be ensured during both training and forward problems using noisy transient training data, thus leading to the development of mass constrained[168] and mass-energy constrained[169] neural network models (i.e., mass constrained neural network or MCNN and mass-energy constrained neural network or MECNN, respectively) for nonlinear dynamic distributed chemical process systems.

In addition to mass and energy conservation, it is desired that the physics-constrained data-driven models also ensure the exact satisfaction of specific thermodynamics constraints while modeling nonlinear chemical process systems, especially in an interconnected[246–249] framework. Interconnected systems in chemical engineering refer to systems that include various integrated chemical processes and unit operations that interact, such as a combined network of reactors, separators, heat exchangers, etc., to achieve the desired outcomes from the systems[250]. Development of accurate physics-based mechanistic first-principles models for such interconnected systems would require a rigorous understanding of each of the involved processes/unit operations followed by the development of the models for each of them separately[251–253]. This would not only consume considerable resources (time and money) but can be difficult to develop for many systems especially novel systems due to lack of clear understanding of all physical/chemical/mechanistic phenomena in such systems. One alternative would be to develop mass and energy constrained data-driven models, such as our previous works [168,169], by using only the input / output boundary data as well as certain thermo-physical properties. However, satisfying thermodynamics constraints are also desired. The purpose of this work is to extend our previous work by including thermodynamics constraints to develop mass, energy, and thermodynamics constrained neural networks (METCNNs).

One difficulty in consideration of thermodynamic constraints in development of data-driven models is to decide/specify what thermodynamic relationship should be considered (such as Raoult's Law or Henry's Law for vapor liquid equilibrium), what model should be used for the desired thermodynamic and thermo-physical quantities (such as fugacity, activity coefficient, enthalpy, vapor pressure, etc.). As opposed to our previous works that exactly satisfied mass and energy constraints simply considering inlet and outlet boundary considerations (under certain assumptions of no mass/energy loss/gain from/to the system), thermodynamic relations and models generally require information about local variables (such as temperature, pressure, etc.) that may not be measured. Thus, for developing METCNN models for interconnected systems, it is imperative that for the unit operations where thermodynamics constraints are desired to be satisfied, measurement of some specific variables are necessary. It should be noted that this aspect is fundamentally different than the first-principles models, where thermodynamics constraints are included for all units where desired.

While only a few papers in recent years have considered thermodynamics constraints during development of optimal data-driven models[239,240,254–256] for chemical systems, no work exists in open literature that addresses transient data with uncertainties using thermodynamically constrained network models and / or considers applying such constrained models to interconnected chemical engineering systems. Ihunde et al.[240] have developed PINN models for two-phase flash vessels, where thermodynamic equations are approximately satisfied by including them as additional penalty terms in the objective function during training. More recently, Idrissi et al.[254] have formulated multiscale thermodynamics-informed NNs (MuTINN) for simulations of large-scale composite structures. However, PINN models built by including constraint equations as penalty terms do not necessarily guarantee the exact satisfaction of the constraint equations during training and can lead to significantly inaccurate results especially during model

simulation. To the best of our knowledge, the only work available in the current literature for exactly satisfying certain thermodynamics equations by the data-driven models is that of Carranza-Abaid and Jakobsen[239]. However, in that work, a first-principles model is constructed along with a surrogate NN model followed by equation decomposition of the physics-based equations. As discussed before, developing the first-principles model can be resource-consuming and difficult for many systems. Furthermore, the approaches proposed for equation decomposition can also be challenging for any generic chemical system. To the best of our knowledge, there is currently no work in the area of NN models for dynamic systems that can exactly satisfy mass, energy, and thermodynamic constraints during training and simulation problems by using the transient data with uncertainties. This work seeks to fill these gaps in the literature on data-driven model building.

It is desired that the mass, energy, and thermodynamics constraints are formulated so that they can be sufficiently expressed without requiring a rigorous understanding of the underlying processes (except for certain thermodynamic models) to formulate the algebraic / differential constraint equations. Furthermore, for data-driven model building, it may not be obvious what thermodynamic constraints must be satisfied due to lack of understanding of unit operation or a group of unit operations across which the measurements are available. Thus, in this work, we consider that where and what thermodynamic constraints should be satisfied are fully user provided and the desired local measurements are available. Under these considerations, an algorithm is developed to represent and include such constraints in data-driven model building. In addition, if the user is unsure about the appropriate thermodynamic model(s) for the system, then the proposed algorithm for optimal METCNN models also downselects the most accurate thermodynamic model for the system through an integer programming based outer layer optimization from the candidate thermodynamics models. The proposed model structures and algorithms have been evaluated for an interconnected system for production of acetic anhydride from vapor-phase acetone.

In summary, this work addresses the following questions in the context of developing data-driven models that can exactly satisfy mass, energy and thermodynamics constraints:

- How to formulate and satisfy mass, energy, and thermodynamics constraints for interconnected chemical systems in a fully data-driven approach? What constraints can be included during training and simulation without necessarily requiring a rigorous understanding of the underlying process?
- How to determine the best thermodynamics model out of all available candidates for a particular nonlinear transient system under consideration under the assumption that desired operational data are available?
- How to ensure exact constraint satisfaction during both training (inverse) as well as, more importantly, simulation (forward) problems?
- How does the presence of uncertainties (noise / bias) in training data affect the training of the physics-constrained models and how to enhance the algorithms to take into account the presence of noise?
- How should the algorithms be developed for both steady-state and dynamic systems? For dynamic data, how can the model structures be exploited to ensure faster computation?

4.3.2 Training and Forward Problem for Steady-State / Dynamic METCNN

This section presents an overview of the formulation of training and forward (simulation) problems for METCNNs during both steady-state and dynamic implementations subject to biased noisy training data. For brevity, only a specific case of error characterization is considered in this report where the training data are generated by adding a time-invariant (constant) bias in presence of an additional Gaussian noise with known variance to the 'true data', i.e., the data obtained from simulating a first-principles model. It is to be noted that the proposed data-driven approaches for METCNN models do not require the development of any first-principles model, and the training / simulation algorithms for METCNNs considered in this work also do not involve the simulated 'true data' ($\boldsymbol{y_{true}}$). However, the simulated data from the first-principles model are valuable for evaluating the performance of the proposed algorithms since the 'truth' is known and are thus useful for comparing the relative performances of the unconstrained (i.e., NN model without any mass, energy, or thermodynamics constraints) and constrained (i.e., METCNN) models to the system truth. In addition, performance of the algorithms for identifying the best thermodynamics model for a particular system can also be evaluated if the truth is known. The training / target data ($\boldsymbol{y_{train}}$) for the implementation of both steady-state and dynamic METCNN models have been generated as:

$$\boldsymbol{y_{train}}(t) = \boldsymbol{y_{true}}(t) + \boldsymbol{\xi} + N(0, \sigma^2) \tag{4.16}$$

where, t denotes the time (for dynamic modeling) or steady-state observation indices (for steady-state modeling) under consideration, $\boldsymbol{\xi}$ represents the time-invariant (constant) bias, and $\mathrm{N}(0, \sigma^2)$ refers to the zero-mean Gaussian noise with known variance (σ^2). The presence of noise in training data can corrupt derivative calculations during parameter estimation and thus, result in biased estimates during training. The primary objective for adding external noise / bias characteristics to simulated data to generate training data is to evaluate the efficiency of the proposed METCNN models in conserving system physics, even when the training data used for estimation of optimal parameters violate the same. Obviously, the proposed algorithms remain consistent for the other types of measurement noise present in data and have demonstrated similar performance in terms of relative prediction accuracy and computational expense for both steady-state and dynamic modeling, as compared to the results obtained by implementing MCNNs and MECNNs separately as can be found in our previous publications[168–170].

Training Problem for Steady-State Modeling

Fig. 4.8 shows the generic block-oriented configuration for formulating the training problem of steady-state METCNN models. As observed from Fig. 4.8, the parameter estimation (i.e., training) problem for steady-state implementation of METCNN models involve the simultaneous regression of parameters (i.e., connection weights and biases) of a conventional feedforward NN (FFNN), along with reconciliation of the outputs obtained from the NN model through steady-state data reconciliation (SDR), thus leading to equality constrained optimization problem given by Eq. (4.17a).

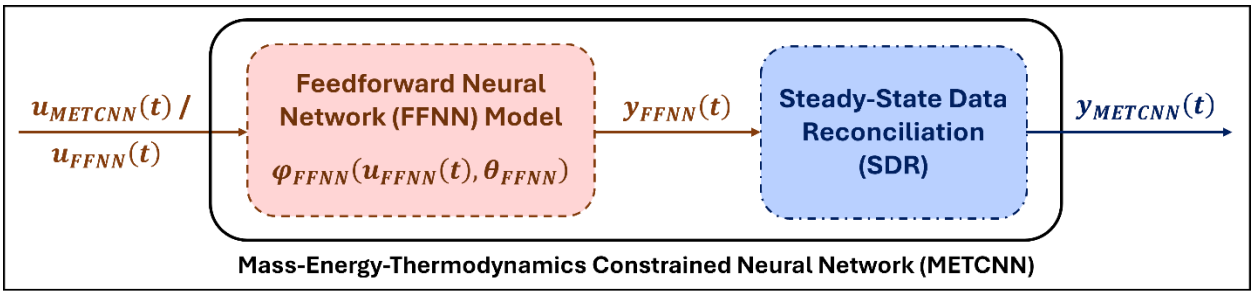


Fig. 4.8. Generic block-oriented configuration for steady-state METCNN models

It may be mentioned here that for this case, the reconciliation of NN outputs itself requires the solution of an optimization problem, and thus has been appended in the objective function (given by Eq. (4.17a)) represented by the second residual term. However, the proposed formulation can readily accommodate any (linear / nonlinear) close-form system of transformation equations in terms of the NN outputs for ensuring the exact satisfaction of the physics conservation laws pertinent to a system under consideration. The mass, energy, and thermodynamics constraints have been imposed as hard equality constraints as represented by $h_M(\cdot), h_E(\cdot)$, and $h_{T_k}(\cdot)$ respectively in Eq. (4.17a). – Eq. (4.17a). During both training and forward problems, all equality (and/or inequality constraints, if any) are incorporated as part of the SDR step, that also provides the necessary degrees of freedom required for solving the equality constrained training problem simultaneously. The outputs from the optimal METCNN model are referred to as $\boldsymbol{y}_{METCNN}(t)$, the intermediate outputs from the FFNN model before reconciliation are denoted as $\boldsymbol{y}_{FFNN}(t)$, while the corresponding model input variables are represented interchangeably as $\boldsymbol{u}_{FFNN}(t)$ or $\boldsymbol{u}_{METCNN}(t)$, where t denotes the steady-state observation indices. The parameters of the FFNN model are collectively referred to as θ_{FFNN} and R denotes the weightage term representing measurement uncertainties.

$$\min_{\boldsymbol{y}_{METCNN}, \theta_{FFNN}} \sum \left(\left(\boldsymbol{y}_{train}(t) - \boldsymbol{y}_{METCNN}(t) \right)^T R^{-1} \left(\boldsymbol{y}_{train}(t) - \boldsymbol{y}_{METCNN}(t) \right) \right) + \sum \left(\left(\boldsymbol{y}_{METCNN}(t) - \boldsymbol{y}_{FFNN}(t) \right)^T R^{-1} \left(\boldsymbol{y}_{METCNN}(t) - \boldsymbol{y}_{FFNN}(t) \right) \right) \tag{4.17a}$$

$$s.t.\ \boldsymbol{y}_{FFNN}(t) = \boldsymbol{\varphi_{FFNN}}(\boldsymbol{u}_{FFNN}(t), \theta_{FFNN}) \tag{4.17b}$$

$$h_M\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \tag{4.17c}$$

$$h_E\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \tag{4.17d}$$

$$h_{T_1}\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \ \vee\ h_{T_2}\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \ \vee\ \ldots. \ \vee\ h_{T_k}\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \tag{4.17e}$$

It can also be noted that the proposed training problem formulation for METCNNs also accommodate the decision for selection of the best thermodynamics model for the system (with k referring to the index of the true thermodynamics candidate model), as evident from Eq. (4.17a). The best thermodynamics model is chosen as the one among different candidate models that yields the most superior predictive performance with respect to the training data (i.e., $\boldsymbol{y_{train}}(t)$) with minimum bias in the sense of the user. It should be noted that the training problem for METCNNs does not include the estimation of parameters for the thermodynamic model (i.e., the model parameters for each candidate thermodynamic model remain constant at their literature values), since the training data are not likely to have sufficient richness for estimation of the parameters of the thermodynamic models and because the training data are likely affected by the equipment performance (i.e., reaction kinetics, mass transfer and heat transfer rates, etc.) along with thermodynamic limitations making it difficult to separate out the relevant data for estimating thermodynamic model parameters. It should also be noted that selection of the best thermodynamic model is not necessary for training METCNNs. If the best thermodynamic model for a system is known a priori, then only that model can be included as the constraint. The specific forms of mass, energy, and especially, thermodynamics constraints would vary from system to system as well as with respect to the types of measurements available depending on the different modes of mass and heat transfer involved, that have been discussed in successive sections. It is our observation that most commercial deep learning packages / libraries such as TensorFlow (v2.14.0), PyTorch (v2.2.2), or NeuroMANCER[257] (v1.4.2) are either not flexible to incorporate equality constraints in machine learning (ML) models (similar to those represented by Eq. (4.17a). – Eq. (4.17a)) or do not allow simultaneous modification of network structure during solving the equality constrained training problem. Therefore, the proposed algorithms are coded in either MATLAB or Python and the optimization problem is solved by using the Interior Point OPTimization[258] (IPOPT) solver available in the OPTimization Interface (OPTI) toolbox in MATLAB (R2021a) as well as using the algebraic modeling language Pyomo[259] (v6.7.1) in Python, respectively. Additional formulations for the training problem subject to varying types of uncertainty characterizations in data can be found in our previous publications on MCNN[168] and MECNN[169].

Forward Problem for Steady-State Modeling

While the optimal parameters obtained by solving the training problem for steady-state METCNN models guarantee the exact satisfaction of mass, energy, and thermodynamics constraints for the entire range / duration of training data, the same parameters may not necessarily lead to exact satisfaction of these physics conservation laws during the forward problem for different input variables other than those used during training. Therefore, the predictions obtained from the optimal FFNN model must still be transformed through a SDR post-processing step, similar to the block-oriented configuration shown in Fig. 4.8, to guarantee the exact conservation of system physics. Unlike typical validation / simulation of conventional NN models, the forward problem for METCNN models will thus also involve solving an equality constrained optimization problem, as expressed in Eqs.(4.18a) - (4.18a). However, it is worth mentioning that the data reconciliation (i.e., SDR) post-processing step in the forward problem formulation is typically associated with much faster computation as compared to model training, since the optimal FFNN model predictions are generally expected to remain sufficiently close to the system truth, i.e., approximately satisfy the physics conservation constraints. In Eqs.(4.18a) - (4.18a), $\theta_{FFNN_{opt}}$ represented the optimal FFNN parameters obtained during model training and $y_{FFNN_{opt}}(t)$ denotes the intermediate outputs obtained from the optimal FFNN model before passing through the reconciliation post-processing step during the forward problem for METCNNs.

$$\min_{\boldsymbol{y}_{METCNN},\theta_{FFNN}} \sum \left(\left(\boldsymbol{y}_{train}(t) - \boldsymbol{y}_{METCNN}(t)\right)^T R^{-1} \left(\boldsymbol{y}_{train}(t) - \boldsymbol{y}_{METCNN}(t)\right) \right) + \sum \left(\left(\boldsymbol{y}_{METCNN}(t) - \boldsymbol{y}_{FFNN}(t)\right)^T R^{-1} \left(\boldsymbol{y}_{METCNN}(t) - \boldsymbol{y}_{FFNN}(t)\right) \right) \tag{4.18a}$$

$$s.t.\ \boldsymbol{y}_{FFNN_{opt}}(t) = \boldsymbol{\varphi_{FFNN}}(\boldsymbol{u}_{FFNN}(t), \theta_{FFNN_{opt}}) \tag{4.18b}$$

$$h_M\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \tag{4.18c}$$

$$h_E\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \tag{4.18d}$$

$$h_{T_1}\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \ \vee\ h_{T_2}\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \ \vee \ldots \vee\ h_{T_k}\left(\boldsymbol{u}_{METCNN}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \tag{4.18e}$$

Training Problem for Dynamic Modeling

The steady-state METCNN formulations can further be extended for dynamic modeling implementations by accommodating the entire time-series training data that can be noisy and biased. One of the primary differences between steady-state and dynamic modeling is the corresponding implementation depending on the availability of system holdup information. Sufficient information about accumulation of mass and / or energy within a chemical process system may not often be available during development of data-driven models, especially for industrial applications. However, the lack of holdup measurements for a particular chemical system does not necessarily imply that the developed METCNNs cannot be applied for modeling such systems, but rather that the mass, energy, and thermodynamics constraints can only be imposed (i.e., satisfied) during the steady-state zones in the overall dynamic data during model training. Nevertheless, there may exist certain thermodynamics constraints (for example, vapor-liquid equilibrium (VLE) constraints) that may be imposed always at every time step irrespective of availability of mass / energy holdup information.

Furthermore, it has typically been observed that the conventional FFNN models may frequently lead to ineffective performance for complex transient process systems[166,180]. In one of our previous publications[166], hybrid static-dynamic NN model architectures have been proposed that have shown superior predictive accuracy as compared to several state-of-the-art recurrent neural network (RNN) models, as well as standalone feedforward and recurrent NN structures, especially for large-sized nonlinear dynamic training data. The hybrid model architectures consist of connecting the nonlinear feedforward (FFNN) and nonlinear recurrent (RNN) structures in series and parallel configurations, as can be seen in the respective block-oriented configurations in Fig. 4.1 in Section 4.2. Additionally, the sequential parameter estimation algorithms[166] developed for the hybrid all-nonlinear static-dynamic models facilitate the independent training of the individual static and dynamic NN models, along with an outer level optimization approach to ensure overall convergence and faster computation. Additional details on the model architectures and sequential training algorithms can be found in our previous paper[166]. When the different candidate unconstrained series and parallel hybrid static-dynamic models were implemented for modeling the transient nonlinear training data with respect to the different case study examples considered in this work, the hybrid parallel architecture, consisting of the FFNN and RNN

connected in parallel to each other (as shown in Fig. 4.1 (c) in Section 4.2), performed slightly superior than both types of series models in terms of prediction accuracy as well as computational expense. The relative difference in predictive performances among various candidate series and parallel hybrid all-nonlinear NN models primarily stems from their contrasting architectures as well as the corresponding sequential parameter estimation algorithms discussed in our previous publication[166]. Therefore, the most optimal hybrid model (i.e., the FFNN connected in parallel to RNN type of configuration in this case) has been chosen for implementation during METCNN formulations.

However, the formulation of the training problem for dynamic METCNNs may slightly change depending on the availability of sufficient holdup information (measurements). Fig. 4.9 (a) and Fig. 4.9 (b) show the structural representations of the dynamic METCNN models in absence and presence of holdup measurements respectively, motivated by the hybrid parallel all-nonlinear NN architecture as shown in Fig. 4.1 (c) in Section 4.2. In the absence of system holdup measurements, the mass, energy, and thermodynamics constraints can only be imposed during steady-state conditions. For such cases, it has been proposed that the entire time-series transient training data be distinctly partitioned into steady-state and dynamic zones. The distinctive partitioning of training data provides sufficient flexibility of sequentially training a steady-state METCNN model for the steady-state datapoints, followed by developing a nonlinear dynamic residual model to capture the deviations of the transient time-series training data from the successive steady-states, as shown in Fig. 4.9 (a). Furthermore, it is to be mentioned here that the overall inputs to the hybrid dynamic METCNN models referred to as $\boldsymbol{u_{par}}(t)$ in both Fig. 4.9 (a) and Fig. 4.9 (b) essentially denote the union of the individual inputs to the FFNN model (i.e., $\boldsymbol{u_{FFNN}}(t)$) and the nonlinear dynamic deviation RNN model (i.e., $\boldsymbol{u_{RNN}}(t)$) represented by the nonlinear auto-regressive network with exogenous inputs (NARX-type RNN). Furthermore, in Fig. 4.9 (a), the outputs obtained from the steady-state METCNN and NARX-type RNN models, represented by $\boldsymbol{y_{METCNN}}(t)$ and $\boldsymbol{y_{RNN}}(t)$ respectively, are added to generate the overall outputs (i.e., $\boldsymbol{y_{par}}(t)$) of the dynamic (hybrid parallel) METCNN model in absence of holdup information. The corresponding training algorithm has been shown in **Fig. C. 4** in **Appendix C**.

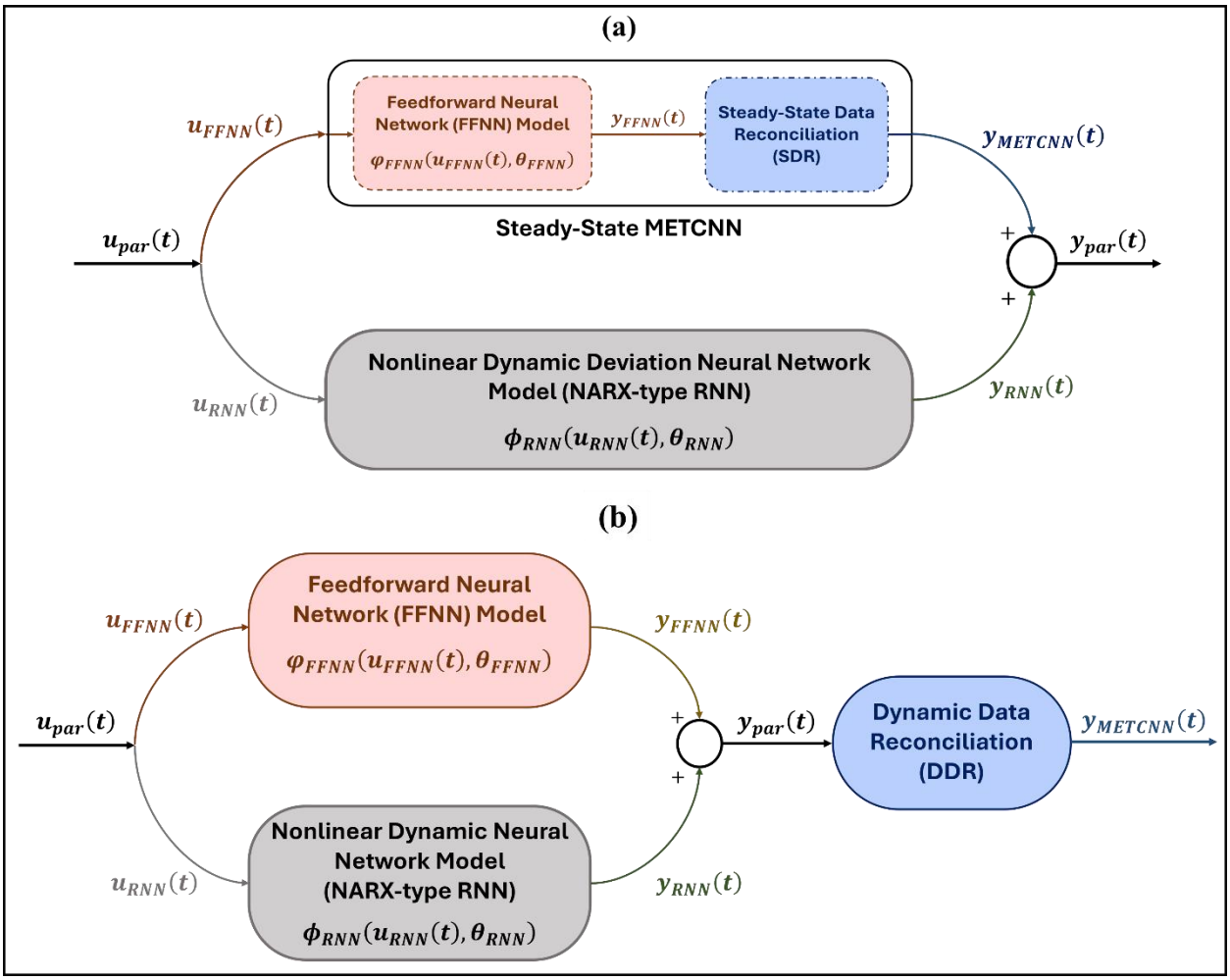


Fig. 4.9. Generic architectures of dynamic (hybrid parallel) METCNN models in (a) absence and (b) presence of system holdup measurements

However, with the availability of holdup measurements, the structure of the hybrid dynamic METCNN model can be slightly modified, as shown in Fig. 4.9 (b). It must be noted here that this case also includes the training / forward problem formulations for dynamic METCNNs when specific thermodynamic constraints (such as VLE constraints, etc.) need to be exactly satisfied at all time steps, even in the absence of sufficient additional information for mass and / or energy holdup within the system. For all such cases, the intermediate outputs obtained from the hybrid all-nonlinear parallel static-dynamic NN model (i.e., $\boldsymbol{y_{par}}(t)$) are passed through a dynamic data reconciliation (DDR) block replacing the SDR step (similar to steady-state modeling) to ensure exact conservation of mass, energy, and thermodynamics constraints at all time steps under consideration. The corresponding formulation of the training problem for the model architecture shown in Fig. 4.9 (b), in presence of mass and energy holdup measurements can be expressed in Eqs.(4.19a)– (4.19a), where t represents the time step (indices) of the transient training data. The mass and energy holdup information for the system has been collectively referred to as $\frac{dM}{dt}$ and $\frac{dE}{dt}$ in Eqs.(4.19a) and (4.19a) respectively. Obviously, for the case where specific thermodynamics constraints (e.g., VLE constraints, etc.) need to be exactly satisfied at all time steps in the absence of necessary mass and/or energy holdup measurements, the respective holdup terms $\left(\frac{dM}{dt}, \frac{dE}{dt}\right)$ can be appropriately dropped while

solving the training problem for dynamic METCNNs. Here, θ_{RNN} denotes the parameters (weights and biases) of the RNN model considered in the hybrid parallel architecture, and $\boldsymbol{\varphi_{FFNN}}(\cdot)$ and $\boldsymbol{\phi_{RNN}}(\cdot)$ signify the system of model equations for the corresponding feedforward and recurrent networks respectively.

$$\min_{\boldsymbol{y}_{METCNN},\theta_{FFNN},\theta_{RNN}} \sum \left(\left(\boldsymbol{y}_{train}(t) - \boldsymbol{y}_{METCNN}(t)\right)^T R^{-1} \left(\boldsymbol{y}_{train}(t) - \boldsymbol{y}_{METCNN}(t)\right) \right) + \sum \left(\left(\boldsymbol{y}_{METCNN}(t) - \boldsymbol{y}_{par}(t)\right)^T R^{-1} \left(\boldsymbol{y}_{METCNN}(t) - \boldsymbol{y}_{par}(t)\right) \right) \tag{4.19a}$$

$$s.t.\ \boldsymbol{y_{par}}(t) = \boldsymbol{\varphi_{FFNN}}(\boldsymbol{u_{FFNN}}(t), \theta_{FFNN}) + \boldsymbol{\phi_{RNN}}(\boldsymbol{u_{RNN}}(t), \theta_{RNN}) \tag{4.19b}$$

$$\frac{dM}{dt} - h_M\left(\boldsymbol{u_{par}}(t), \boldsymbol{y_{METCNN}}(t)\right) = 0 \tag{4.19c}$$

$$\frac{dE}{dt} - h_E\left(\boldsymbol{u_{par}}(t), \boldsymbol{y_{METCNN}}(t)\right) = 0 \tag{4.19d}$$

$$h_{T_1}\left(\boldsymbol{u_{par}}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \ \vee\ h_{T_2}\left(\boldsymbol{u_{par}}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \ \vee \ldots. \ \vee\ h_{T_k}\left(\boldsymbol{u_{par}}(t), \boldsymbol{y}_{METCNN}(t)\right) = 0 \tag{4.19e}$$

The estimation of optimal network parameters for both FFNN and RNN models simultaneously in a monolithic training approach while solving Eqs.(4.19a)– (4.19a) may lead to significantly higher computational expense. To tackle this issue, the sequential training algorithms developed for the hybrid all-nonlinear series and parallel network structures in one of the previous publications by the same authors[166] have been leveraged. Such decomposition-based approaches ensure relatively faster computation by decomposing the hybrid model into separate individual static (FFNN) and dynamic (RNN) networks followed by training each of them independently. However, it must be mentioned here that since the optimization problem formulated for dynamic METCNNs in presence of holdup information (as represented in Eqs.(4.19a)– (4.19a)) considers the entire time-series data within the objective function, the associated computational expense during model training is significantly higher as compared to the training of dynamic METCNN models in absence of system holdup measurements that only considers the steady-state data during constrained optimization, thus representing a much smaller subset of the entire dynamic training dataset. Additional alternative formulations of the training problems for such dynamic physics-constrained NN models can be found in our previous publications[168,169].

Forward Problem for Dynamic Modeling

The formulation of the forward problem for dynamic METCNN models follow similar order / sequence of approaches like the corresponding training problem. The estimates obtained from the hybrid all-nonlinear series static-dynamic NN structure are passed through a DDR / SDR post-processing step which ensure the exact conservation of mass, energy, and thermodynamics constraints for each dynamic time step (if holdup measurements are available) or at every steady-state zone identified within the entire transient duration (if holdup measurements are not available), respectively. Similar to the training problem, alternative formulations of the forward problems for dynamic implementations of such physics-constrained data-driven models (e.g., MCNN, MECNN, METCNN) can be found in our previous publications[168,169].

4.3.3 Case Studies and Corresponding Mass Energy Thermodynamics Constraints

The proposed METCNN architectures and training / simulation algorithms discussed above are evaluated for various nonlinear chemical process systems. While for most chemical processes, the quantification of mass conservation is unambiguous and straightforward, quantifying the overall energy balance of the system in presence of appropriate thermodynamics constraints may be relatively challenging in a fully data-driven approach due to different possible sources of energy transfers associated with a system including (but not limited to) generation / depletion of heat as well as energy transformation from one form to the other within the system. Although the MCNN, MECNN, and METCNN models developed in this research have been implemented to develop data-driven physics-constrained models for several nonlinear dynamic chemical processes, only the results obtained for a lumped / distributed adiabatic superheater system as well as for an interconnected system considering the production of acetic anhydride have been discussed here for brevity. Additional implementations of the different types of physics-constrained NN models can be found in the corresponding papers[168,169] published as part of this project.

For all case studies considered in this work[170], the training and validation (simulation) data have been generated from respective first-principles models subject to step-changes in model input before contaminating them with different types of noise / bias characterizations. It is also assumed that we have access to measurements for all input and output variables necessary to ensure mass and energy conservation. The input and output variables are normalized using min-max normalization and 60% of the entire dataset are allocated for training the METCNNs, while the remaining 40% is used for model simulation (validation). We also perform k-fold cross validation (with k = 5) for all case studies. Both the standard logistic sigmoid and hyperbolic tangent activation functions have been used in the neural network models considered in this work. Even though the NNs are trained with respect to the normalized values of the input and output variables, it is ensured that the mass and energy balance constraints involve the actual absolute values of the variables concerned. For steady-state modeling, the initial guess for the METCNN parameters is determined by training a standalone FFNN without any mass balance constraints (i.e., without the SDR block) irrespective of the noise characterization. All neural network models considered in this work are structured as fully connected single hidden layered networks with the number of neurons in the input and output layers exactly equal to the number of input and output variables, respectively.

Case Study 1: Adiabatic Superheater / Reheater System

The development of accurate data-driven models for superheater / reheater systems poses to be significantly challenging, due to their inherent nonlinearities and complexity. This work focuses on developing both lumped and distributed data-driven models of a superheater / reheater system, under the following assumptions:

- The process is adiabatic, i.e., no heat depletion / loss / generation within the system.
- No transformation of energy takes place from one form to the other (such as due to resistive heating, viscous heat dissipation, etc.) within the system.
- No change of phase occurs among the different species involved in the system.

A schematic of the lumped parameter model for the superheater / reheater system is shown in Fig. 4.10(a), where flue gas and steam flows in a crossflow arrangement. Since the overall mass of both steam and flue gas are trivially conserved in such a system, the mass flowrates of the two streams $(\dot{m}_{St}, \dot{m}_{FG})$ are considered to be exactly the same at the inlet and outlet of the superheater / reheater system. No spatial variation is considered for steam and flue gas flows while developing the lumped parameter model[260]. For brevity, the system thermodynamics constraints have not been considered for this system. For both lumped and distributed models, the input variables of the MECNN are represented by the inlet mass flowrates, inlet temperatures $\left(T_{St,in}, T_{FG,in}\right)$ and pressures (P_{St}, P_{FG}) of steam and flue gas streams, whereas the outlet temperatures $\left(T_{St,out}, T_{FG,out}\right)$ of both streams have been considered as model outputs. Furthermore, since only energy conservation laws need to be incorporated for this case study, the optimal METCNNs thus obtained will be simply referred to as Energy Constrained Neural Networks (ECNNs).

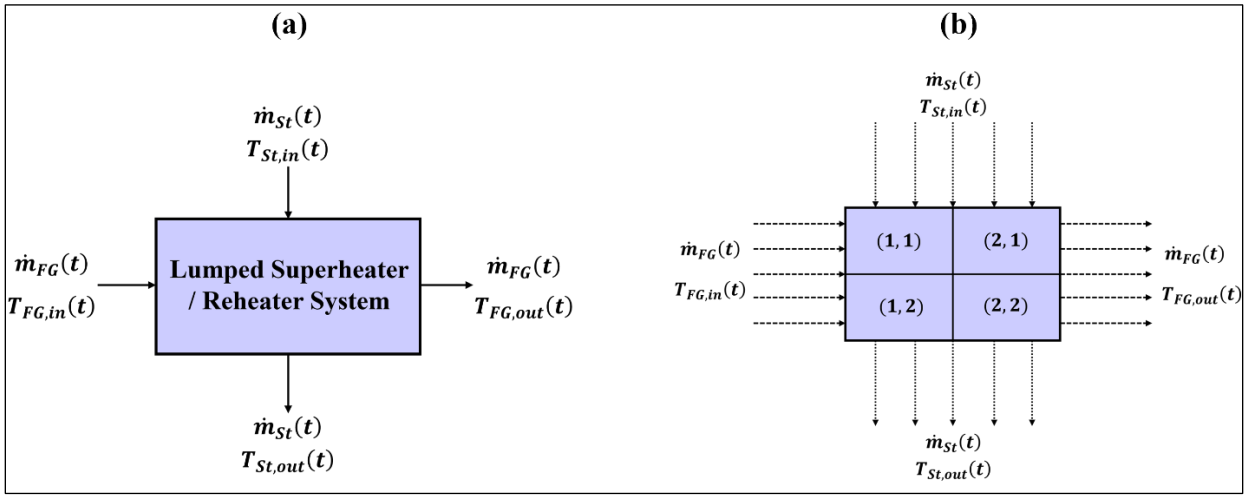


Fig. 4.10. Schematic of the (a) lumped and (b) distributed superheater / reheater model for ECNN implementation

The energy balance constraints $\left(\boldsymbol{h_E}(\cdot)\right)$ imposed during formulation of both training and forward problems for the lumped parameter ECNN model can be expressed as steady-state enthalpy balance equations involving the input and output boundary conditions, as given in Eq. (4.20).The specific heat capacity of the flue gas stream $\left(C_{p_{FG}}\right)$ is assumed to be a constant value calculated from ideal gas laws, whereas the specific enthalpies of steam $(h_{St,in}, h_{St,out})$ have been calculated using functions defined in the IAPWS R7-97 formulation for the properties or water and steam in terms of steam inlet / outlet temperature and pressure. It should be noted that the algorithm is agnostic to the specific formulation of the energy conservation constraint.

$$\dot{m}_{FG}(k)\, C_{p_{FG}}(k)\, \left(T_{FG,in}(k) - T_{FG,out}(k)\right) = \dot{m}_{St}(k)\, \left(h_{St,out}\left(T_{St,out}, P_{St}, k\right) - h_{St,in}\left(T_{St,in}, P_{St}, k\right)\right) \tag{4.20}$$

Though the lumped ECNN model ensures that the energy conservation laws are exactly satisfied at the boundary conditions during both training and forward problems, no information can be derived regarding the distribution of temperature within the superheater / reheater system. Therefore, the proposed algorithms for developing the lumped parameter model can be easily extended to construct a distributed model, if needed, by imposing additional constraints on the ECNN model representing the closure of energy balance for each discrete element of the spatial coordinates. The schematic of the distributed

superheater / reheater model for ECNN implementation has been shown in Fig. 4.10 (b). For simplicity, the superheater / reheater model, as shown in Fig. 4.10 (a), has been discretized into two control volumes each in the horizontal and vertical directions, thus leading to a 2 x 2 grid structure as shown in Fig. 4.10 (b). Obviously, a larger number of discretization grid points may be considered for higher spatial resolution, if needed[147]. The flow pattern as well as the input and output variables to the ECNN model remains the same for both flue gas and steam as compared to the lumped model. However, for both training and forward problems while developing ECNN for the distributed model, the enthalpy balance constraint equation (Eq. (4.20)) has to be exactly satisfied for each of the individual control volumes in spatial directions, along with the boundaries of the system. Local measurements of both flue gas and steam temperatures at the intermediate grid points have been considered during development of optimal ECNNs.

Case Study 2: Interconnected System for Production of Acetic Anhydride

The performance of the proposed METCNN models is evaluated by modeling the production of acetic anhydride from acetone as a representative nonlinear interconnected chemical system [261–263]. Fig. 4.11 shows the process flowsheet for the production of acetic anhydride from acetone (CH_3COCH_3). The development of accurate predictive models for this overall system is desired for optimizing the profitability and economics associated with the manufacturing process. Vapor-phase acetone (A) is preheated in a heat exchanger (heater) before undergoing cracking in a tubular plug flow reactor (PFR) to produce methane (CH_4) (B) and ketene (CH_2CO) (C). The product mixture is then passed through a series of coolers before entering the flash separator. In the flash separator that operates under a constant pressure and temperature, the gas phase separates from the liquid phase, followed by recycling the bottom (liquid) stream through a heater back to the gaseous feed stream for the PFR to utilize the unreacted acetone. The ketene from the bottom product of the flash separator finally reacts with acetic acid (CH_3COOH) in a CSTR to produce acetic anhydride. It should be noted that, if necessary, boundary information (measurements) is available for all process systems / unit operations shown in Fig. 4.11, then they can be separately modeled by MCNN / MECNN by using appropriate constraint formulations. For example, the CSTR system considering the reaction of acetic acid and ketene to form acetic anhydride can be modeled by an MECNN if the inlet and outlet concentration and temperature measurements are available, similar to one of the case studies presented in our previous publication[168]. However, for certain processes within the interconnected system, it may be desired to modify / update/augment the mass and energy conservation constraints with due consideration of thermodynamic constraints. For instance, for thermodynamically consistent modeling of the tubular reactor (PFR) for the vapor phase cracking of acetone (represented by the first red dotted box in Fig. 4.11), the energy balance constraints for METCNNs can be formulated by including additional enthalpy departure formulations based on system thermodynamics expressed in terms of input / output boundary conditions as well as various thermo-physical properties without the requirement of rigorous understanding (knowledge) about the reaction kinetics.

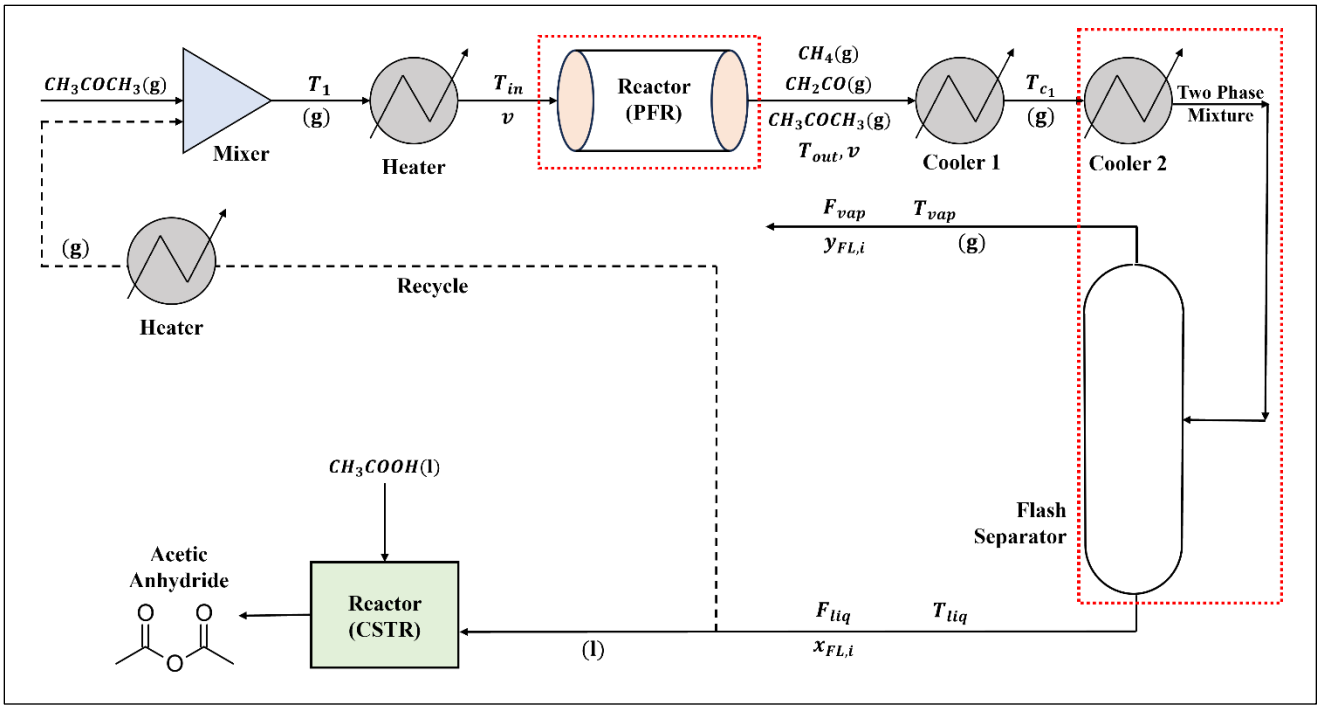


Fig. 4.11: Process flowsheet of acetic anhydride production for METCNN implementation

Furthermore, a portion of the vapor (gas) phase stream entering the final cooler / condenser (heat exchanger), represented as 'Cooler 2' in Fig. 4.11, gets condensed. While the temperature of the exiting stream from 'Cooler 2' can be easily measured, it is difficult (if not impossible) to measure the vapor fraction of that stream and composition of the liquid and vapor fractions. However, the composition and flowrate of liquid and vapor streams from the flash separator can be measured. Therefore, for such systems, the combined system involving complex simultaneous heat transfer and flash separation in the heat exchanger (HX) / cooler and separator respectively, can be collectively considered as a single system (as demonstrated by the second red dotted box in Fig. 4.11) to be modeled by the proposed METCNN algorithms. In summary, this section discusses the performance of METCNN models[264] for both steady-state and dynamic implementations with respect to the tubular (plug flow) reactor as well as the combined HX-flash system. Various forms of potential candidate thermodynamics-based constraint equations have been proposed in addition to mass and energy balances for efficient modeling of interconnected systems. However, it has been assumed that holdup measurements for both of the distributed systems under consideration are not available during the development of optimal data-driven models, and hence the proposed physics conservation laws have only been applied at steady-state following the model architecture and training algorithm, especially during dynamic modeling applications. Obviously, in presence of sufficient holdup measurements, the mass, energy, and thermodynamics constraints can be exactly satisfied at every time step by appropriately modifying the training (and forward) problem based on the model structure and algorithm discussed in Fig. 4.9 (b) and Eqs.(4.19a)– (4.19a) in Section 4.3.2.

- **METCNN Formulation for Plug Flow Reactor (Enthalpy Model)**

The vapor-phase cracking reaction for conversion of acetone (A) to form methane (B) and ketene (C) constitutes the principal step in the manufacture of acetic anhydride in process industries. The detailed mathematical model and nominal parameter values for this endothermic reaction, typically taking place in a PFR, can be found in numerous references[261–263,265]. The reactor feed has been assumed to contain a mixture of acetone (predominant), and small amounts of methane and ketene (present in the recycle) in vapor phase and the volumetric flow rate of the reaction mixture is assumed to remain constant across the tubular reactor. The input variables for the METCNN models are represented by the inlet concentration of acetone $\left(C_{A_{in}}\right)$, methane $\left(C_{B_{in}}\right)$, and ketene $\left(C_{C_{in}}\right)$ in reactor feed, the volumetric flow rate of the reactant / product streams (v), and the inlet temperature of the feed stream (T_{in}), while the model outputs comprise the outlet concentrations of A (C_A), B (C_B), and C (C_C), as well as the temperature of the product stream (T_{out}). While the mass balance constraints imposed during parameter estimation for solving both training (inverse) and forward problems in this case can easily be represented by the elemental atom balance equations expressed in terms of input-output boundary conditions, the energy (enthalpy) balance constraints can be expressed using appropriate thermodynamic models for departure functions to explicitly account for the non-ideality in the underlying process without requiring a rigorous understanding of the system such as reaction kinetics, etc. For reference, with respect to the outputs from the optimal METCNN models (i.e., $C_{A_{METCNN}}, C_{B_{METCNN}}, C_{C_{METCNN}}, T_{METCNN}$), the corresponding mass (atom) balance constraints for C, H, and O atoms are shown in Eqs. (4.21) through (4.23), whereas Eq. (4.24) denotes the overall energy (enthalpy) conservation constraints in terms of inlet / outlet molar flow rates $(\dot{m}_{in}, \dot{m}_{out})$ and molar enthalpies (h_{in}, h_{out}) for the reactor under consideration. In Eqs. (4.21)–(4.23), n_{C_j}, n_{H_j}, and n_{O_j} respectively represent the number of C, H, and O atoms present in the j^{th} species in the feed / product streams.

$$\sum_{j=A,B,C} n_{C_j} * v(t) * C_{j_{in}}(t) = \sum_{j=A,B,C} n_{C_j} * v(t) * C_{j_{METCNN}}(k) \tag{4.21}$$

$$\sum_{j=A,B,C} n_{H_j} * v(t) * C_{j_{in}}(t) = \sum_{j=A,B,C} n_{H_j} * v(t) * C_{j_{METCNN}}(k) \tag{4.22}$$

$$\sum_{j=A,B,C} n_{O_j} * v(t) * C_{j_{in}}(t) = \sum_{j=A,B,C} n_{O_j} * v(t) * C_{j_{METCNN}}(k) \tag{4.23}$$

$$\dot{m}_{in}(t)\, h_{in}(t) - \dot{m}_{out}(t)\, h_{out}(t) = 0 \tag{4.24}$$

The appropriate expressions for the molar enthalpies (h_{in}, h_{out}) in terms of the reactor inlet and outlet temperatures define the thermodynamics constraints imposed on the parameter estimation problem. Typically, for real gases, the molar enthalpy (h) at a particular temperature may be represented as the summation of the respective molar enthalpy of an ideal gas $\left(h^{ig}\right)$ at that same temperature, and the deviations of the system properties from ideal gas behavior denoted by the respective departure functions[266] (h^{ex}), as follows:

$$h(t) = h^{ig}(t) + h^{ex}(t) \tag{4.25}$$

Furthermore, since molar enthalpy is essentially a state function, the enthalpy departure functions can be explicitly expressed in terms of the corresponding inlet and outlet temperatures and compositions, given a certain equation of state (EOS) chosen by the user and its derivatives. The different candidate thermodynamics models, i.e., EOS considered for this example to express $h^{ex}(t)$ are listed in Table 4.5,

along with their respective indices (k) to identify the best thermodynamics model for this system. Appropriate representations to express $h^{ex}(t)$ as explicit functions of inlet and outlet temperatures based on each candidate EOS model can be imposed as additional thermodynamics constraints during formulations of the training and forward problems for METCNN models.

Table 4.5. List of candidate thermodynamics models considered for the PFR system to explicitly express $h^{ex}(t)$ as a function of T_{in} and T_{out}

Index for Candidate Thermodynamics Model (k)	**Equation of State (EOS) considered**
1	Ideal Gas, i.e., $h^{ex}(t) = 0, h(t) = h^{ig}(t)$
2	Peng-Robinson (PR) EOS
3	Virial EOS

It is worth clarifying in this regard that the objective of this research is not to develop an accurate enthalpy model for a particular system under consideration. This work specifically focuses on analyzing what types of thermodynamics constraints may be imposed during development of optimal physics-constrained data-driven models and discusses algorithms for solving the constrained optimization problems during model training and simulation. The specific mathematical form of each of the thermodynamic constraint equations would be user provided. The proposed approach ensures that the outputs obtained from optimal METCNN models are thermodynamically consistent. Furthermore, the particular thermodynamics model that yields the most superior accuracy with respect to the training data (measurements) will be considered as the best model for this system with minimum bias.

- **METCNN Formulation for Combined HX-Flash System (VLE Model)**

Unlike the reactor system considered above, the combined HX-flash process is considered as a single black-box system for the implementation of steady-state and dynamic METCNN models. Since the local temperature and composition measurements are assumed to be unavailable at the outlet of the final heat exchanger before the flash separator (referred to as 'Cooler 2' in Fig. 4.11), the HX and flash separator systems cannot be modeled separately by using appropriate MCNN / MECNN / METCNN models. Therefore, in this case, the input variables for the METCNN model are represented by the inlet concentrations of A $\left(C_{A_{in}}\right)$, B $\left(C_{B_{in}}\right)$, and C $\left(C_{C_{in}}\right)$ entering the HX, along with the corresponding inlet temperature $\left(T_{c_1}\right)$ and volumetric flow rate (v). Similarly, the model output variables are denoted by the measured outputs from the flash separator, namely, the molar flow rate $\left(F_{vap}\right)$, outlet temperature $\left(T_{vap}\right)$, and composition / mole fractions $(y_{FL,i})$ of the vapor phase, as well as the molar flow rate $\left(F_{liq}\right)$, outlet temperature $\left(T_{liq}\right)$ and composition / mole fractions $\left(x_{FL,i}\right)$ of the liquid phase. For the development of accurate physics-constrained data-driven models for such systems, it may not be sufficient to only implement the appropriate mass and energy balance constraints across the system under consideration. For instance, for the flash separator operating under constant temperature and pressure, it becomes

significantly critical to ensure that the predicted vapor phase and liquid phase mole fractions are consistent with the corresponding VLE model, i.e., the K-value remains constant for each species as calculated by the appropriate K-value model. Such thermodynamics-based constraints may not necessarily be satisfied while imposing only the mass or energy balance constraints during both training (inverse) and forward problems.

Similar to Eqs. (4.21)– (4.23), the mass conservation constraints for this case can also be represented in terms of elemental C, H, and O balance equations, albeit using a slightly different mathematical representation in terms of the process input and output variables, as follows:

$$\sum_{i=A,B,C} n_{C_i} * v(t) * C_{i_{in}}(t) = \sum_{i=A,B,C} n_{C_i} * y_{FL,i}(t) * F_{vap}(t) + \sum_{i=A,B,C} n_{C_i} * x_{FL,i}(t) * F_{liq}(t) \quad (4.26)$$

$$\sum_{i=A,B,C} n_{H_i} * v(t) * C_{i_{in}}(t) = \sum_{i=A,B,C} n_{H_i} * y_{FL,i}(t) * F_{vap}(t) + \sum_{i=A,B,C} n_{H_i} * x_{FL,i}(t) * F_{liq}(t) \quad (4.27)$$

$$\sum_{i=A,B,C} n_{O_i} * v(t) * C_{i_{in}}(t) = \sum_{i=A,B,C} n_{O_i} * y_{FL,i}(t) * F_{vap}(t) + \sum_{i=A,B,C} n_{O_i} * x_{FL,i}(t) * F_{liq}(t) \quad (4.28)$$

Furthermore, since the model outputs are represented by individual mole fractions of the species in both liquid and vapor phases, the following compositional constraints are also considered as additional mass conservation constraints:

$$\sum_i y_{FL,i} = 1 \quad (4.29)$$

$$\sum_i x_{FL,i} = 1 \quad (4.30)$$

Similarly, the steady-state energy (enthalpy) balance equations can simply be expressed in terms of the inlet and outlet boundary conditions considering the entire system as follows:

$$F_{in}h_{in} - F_{liq}h_{out_{liq}} - F_{vap}h_{out_{vap}} - \dot{Q}_c = 0 \quad (4.31)$$

where, F_{in} denotes the molar flow rate of inlet stream into the HX-flash system, h_{in}, $h_{out_{liq}}$ and $h_{out_{vap}}$ represent the molar enthalpies of the inlet gas, outlet liquid phase and outlet vapor phase streams respectively, and $\dot{Q}_c$ refers to the rate of heat removal in the heat exchanger (cooler). It is to be noted that the cooler heat duty $(\dot{Q}_c)$ can be expressed in terms of the change of enthalpy of the specific utility under consideration for which utility flowrates and inlet and outlet temperatures are generally measured. Furthermore, the molar enthalpies of the outlet vapor and liquid phases can be related to each other, as given by:

$$h_{out_{vap}} = h_{out_{liq}} + q_{vap}^{ig} + q_{vap}^{ex} \quad (4.32)$$

where, q_{vap}^{ig} and q_{vap}^{ex} respectively refer to the heat (enthalpy) of vaporization for ideal gas solution and the corresponding excess enthalpy (departure function) associated to account for the non-ideality in the process. The departure function model accounts for the additional intermolecular interactions and deviations from ideal behavior that can affect the vaporization process[267]. Appropriate enthalpy departure function models given a specific EOS[266] chosen by the user may be deployed. Additionally, a variety of activity coefficient models can also be considered for representing the liquid-phase non-ideality for predicting phase equilibrium behavior, and liquid enthalpy. The explicit mathematical

representations of such activity coefficient models given a specific temperature and pressure are available in standard thermodynamic text books and have been investigated for specific systems by researchers[268,269] and therefore, are not included here. Finally, the vapor phase and liquid phase compositions of each species involved in the process are constrained by the K-value model to ensure vapor-liquid equilibrium, thus leading to the following thermodynamics constraints for this system under consideration:

$$K_i = \frac{y_{FL,i}}{x_{FL,i}} \qquad \forall\, i = \{A, B, C\} \tag{4.33}$$

As evident from Eq. (4.32), the energy (enthalpy) balance constraints imposed for this system involves the outlet mole fractions (i.e., $x_{FL,i}$) which are, in turn, further constrained by the VLE / K-value equations (as given by Eq. (4.33)) for a given temperature and pressure. Therefore, exact satisfaction of these mass, energy and thermodynamics constraints simultaneously during inverse and forward problems is highly desired while developing accurate data-driven models for such systems. Obviously, the specific form of such constraints may differ based on the specific thermodynamic model. This work is primarily focused on the development of approaches that can be implemented during formulation of physics-based constraints in the proposed METCNN models.

4.3.4 Results and Discussions: Steady-State Modeling

This section discusses the steady-state modeling results obtained from the optimal ECNN / METCNN models for the case study examples under consideration. For brevity, only the results obtained from the implementation of ECNN on the distributed superheater system and those obtained from METCNN implementation on the PFR are only discussed in this report. Additional discussion for the other systems considered for model implementation may be found in our previous publications[168,169,264]. For all results discussed in this report, the results obtained from optimal METCNN models are compared with those obtained from an unconstrained NN model, i.e., in absence of any mass / energy / thermodynamics constraints when both the models have been trained using the same noisy biased training data (referred to as 'Measurements'). The relative model performances in terms of violating the physics constraints imposed during training and forward problems have been quantified in terms of absolute percentage errors (APE) in mass and energy balances, as well as certain thermodynamics constraints (if necessary).

Case Study 1: Adiabatic Superheater / Reheater System

Since mass is trivially conserved for this case study, the performance of ECNNs have only been analyzed based on model efficiency in exactly satisfying the energy conservation of the system. The comparison of error in energy balance for all cases of error characterizations considered for the lumped and distributed ECNN models has been evaluated in terms of APE in energy (Q) balance defined as:

$$APE_{energy} = \left|\frac{Q_{FG} - Q_{St}}{Q_{FG}}\right| * 100\% \tag{4.34}$$

where, Q_{FG} and Q_{St} refer to the heat flux expressions for flue gas and steam respectively.

As discussed before, the development of the distributed model of the superheater / reheater system involves the exact satisfaction of the energy balance constraint equation (as given by Eq. (4.20)) at each discretized element (control volume) in the spatial coordinates, in addition to the system boundaries. In addition to considering that all local measurements are available during development of optimal ECNN models due to optimal placement of measurement sensors in the system[270,271], sensitivity analyses

have also been performed to describe how the unavailability of some local measurements influence the optimal model development. A constant bias equaling to 5 °C is added to each output variable along with an additional Gaussian noise with $\mu = 0$ and $\sigma^2 = 1$. The training results obtained from the distributed ECNN for this case have not been included here for brevity. Fig. 4.12 shows the model results vs truth for the ECNN and NN w/o energy constraints respectively for simulation data, at one of the individual discretized grid points within the distributed system. As expected from Fig. 4.12 (a), it is evident that the outputs of the standalone neural network (w/o energy constraints) show an excellent match (0.2% RMSE) with the measurements while leading to larger errors (around 1.0% RMSE) with respect to the true data. Furthermore, the ECNN also exactly satisfies energy conservation constraints both at the boundary as well as intermediate grid points, whereas the unconstrained NN shows maximum APE in energy balance as high as 5.5% at the boundary and 32% at one of the intermediate grids. The comparison of error between ECNN and NN w/o energy constraints for energy balance at the intermediate discretized grid points in the spatial coordinates can be seen in **Fig. C. 5** in **Appendix C.**. It has been observed that the ECNN does exactly satisfy energy conservation at steady state showing an accurate match with the true data during both training and simulation.

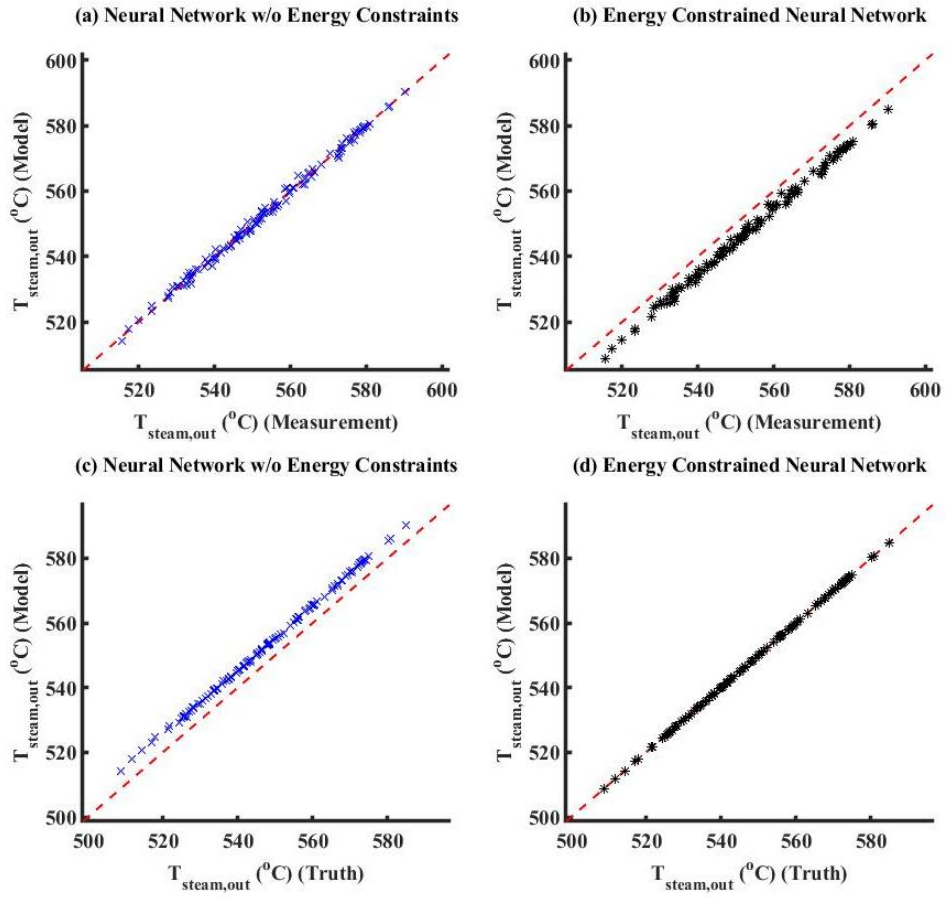


Fig. 4.12: Comparison of results between ECNN and NN w/o energy constraints for the simulation data of $T_{St,out}$

However, in many practical applications, it is highly unlikely that all local states will be measured with reasonable accuracy. Therefore, the value / quality incorporated by such local measurements during development of a distributed data-driven model has been analyzed by performing sensitivity studies. Two types of analyses have been performed where only a fraction of all available local measurements are considered during optimal model synthesis and model simulation. The first approach is based on 'sampling', where multiple subsets of all available local measurements at intermediate discretization grids are extracted and considered for model development, in addition to all boundary conditions. This approach also assumes that measurements are available at all intermediate locations, but not for all steady-state (or dynamic) observation indices as compared to the boundary measurements. On the contrary, the second approach is based on 'availability' of sensor (measurement) data at the individual discretization elements in the spatial coordinates. Unlike the sampling approach, this analysis assumes that local measurements are available for all steady-state (or dynamic) indices but not at all locations. For both sensitivity analyses, the minimum % RMSE calculated between the ECNN model outputs and true data among all possible combinations of data selection is given by Eq. (4.35), where k_n denotes the size of simulation data, n denotes the total number of output variables and $\boldsymbol{C}$ denotes the set of all possible combinations for each sensitivity analysis under consideration. It is also to be noted that there exists a finite number of combinations for the second approach (i.e., when the local measurements are not available at one or more discretization grids during model development). The results presented below are for the local measurement data for which min % RMSE is obtained for a given availability of % local measurement.

$$\min \%\ RMSE = \min \left\{ \sqrt{\frac{1}{n * k_n} \sum_{k=1}^{k_n} \sum_{i=1}^{n} \left(\frac{\boldsymbol{y}_{i,true}(k) - \boldsymbol{y}_{i,ECNN}(k)}{\boldsymbol{y}_{i,true}(k)} \right)^2} \right\}_{c \,\epsilon\, \boldsymbol{C}} \tag{4.35}$$

Fig. 4.13 show the minimum percentage RMSE for simulation data calculated between ECNN outputs and system truth corresponding to the two sensitivity analyses based on availability of local measurements. It can be clearly observed that the minimum RMSE monotonously decreases as more and more local information is made available during optimal model synthesis. Obviously, this type of sensitivity analyses can be performed for the other case studies during both steady-state and dynamic implementations as well under all different types of error characterizations.

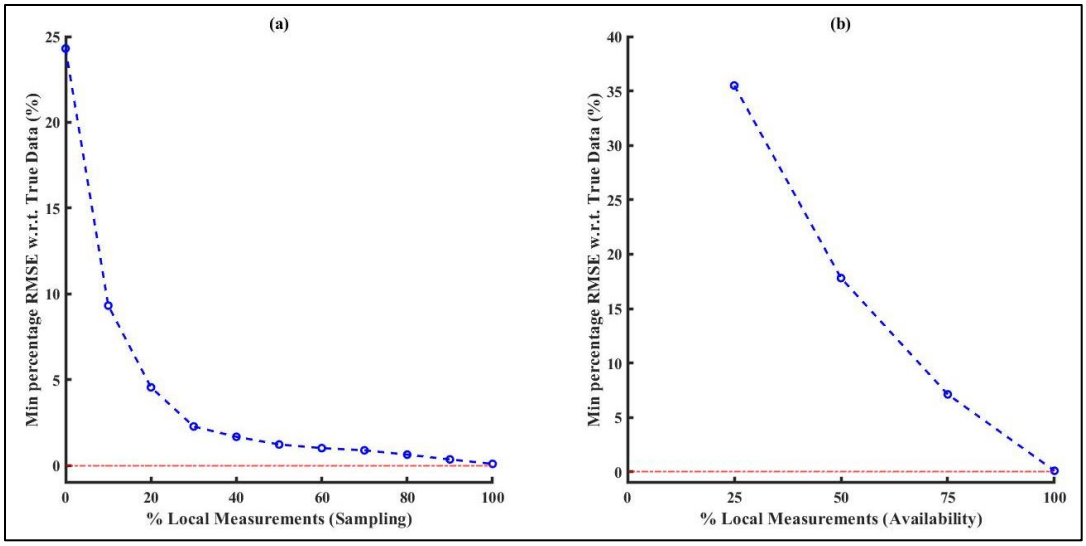


Fig. 4.13. Comparison of minimum % RMSE for simulation data using ECNN from sensitivity analyses based on (a) sampling and (b) availability of local measurements

Case Study 2: Interconnected System for Production of Acetic Anhydride

For reference, it must be mentioned that in all results discussed for this case study, the results obtained from the MECNN model refer to those obtained while considering only the mass (atom) and energy (ideal gas enthalpy) balance constraints, in absence of any additional thermodynamics constraints such as enthalpy model in terms of appropriate departure function for the reactor system. The proposed METCNN models and algorithms have been implemented to model the tubular PFR involving the vapor-phase cracking of acetone, by updating the ideal gas energy balance constraints with additional thermodynamics-based enthalpy model equations, as discussed in Section 4.3.2. For all results discussed in this section, the APE in energy (Q) balance is defined as:

$$APE_{energy} = \left| \frac{\dot{m}_{in} h_{in} - \dot{m}_{out}\, h_{out}}{\dot{m}_{in} h_{in}} \right| * 100\% \tag{4.36}$$

Similarly, the APE in mass balance for the i^{th} atom can be defined as:

$$APE_{atom_i} = \left| \frac{\sum_{in} F_{m_{atom_i}} - \sum_{out} F_{m_{atom_i}}}{\sum_{in} F_{m_{atom_i}}} \right| * 100\% \tag{4.37}$$

where, $\sum_{in} F_{m_{atom_i}}$ and $\sum_{out} F_{m_{atom_i}}$ respectively represent the total mass flowrate of the i^{th} atom in all inlet and outlet streams.

In addition to APE, the predictive performances of the different candidate constrained and unconstrained NN models have been compared to both the system truth (i.e., true data) and measurements (i.e., training data) in terms of another performance metric represented by percentage root mean squared error (RMSE). The percentage RMSE values are calculated between the optimal model predictions and the corresponding measurements / true data to evaluate whether the presence of mass, energy, and especially thermodynamics constraints yield solutions sufficiently close to the system truth, even when trained with

measurements that show significant mismatches with the true data. Furthermore, for this case, since the outer layer integer programming problem consists of only 3 candidate thermodynamics models, the selection of the optimal model is performed through comprehensive enumeration. Alternative integer programming algorithms may also be developed and / or implemented for larger family of candidate functions, if necessary. Table 4.6 shows a summary of performances of the optimal METCNN for comparison with respect to both measurements (training data) and system truth (true data) for T_{out} during training and simulation when different candidate thermodynamics models (as listed in Table 4.5) have been considered as additional constraints. From Table 4.6, it is clearly evident that the optimal METCNN consisting of the PR EOS significantly outperforms the other candidate thermodynamics models (i.e., ideal gas and virial EOS) in terms of the corresponding percentage RMSE values with respect to the training data (measurements) with minimum bias, thus leading to the selection of PR EOS as the best thermodynamics model (i.e., $k_{opt} = 2$) for the system under consideration. Therefore, the remaining results for all steady-state and dynamic implementations of METCNN models on the PFR system within the interconnected flowsheet under consideration correspond to those obtained from using the best thermodynamics (i.e., Peng-Robinson) model for evaluating the enthalpy departure functions, that are imposed as part of the physics conservation constraints during both training and forward problems.

Table 4.6. Comparison among different candidate thermodynamics models imposed as constraints in METCNN in terms of percentage RMSE for training and simulation data of T_{out}

Index for Candidate Thermodynamics Model (k)	**Equations of State (EOS) considered for $h^{ex}(t)$**	**Training RMSE for T_{out} (%) with respect to**		**Simulation RMSE for T_{out} (%) with respect to**	
		Training Data	**True Data**	**Training Data**	**True Data**
1	Ideal Gas, i.e., $h^{ex}(t) = 0$, $h(t) = h^{ig}(t)$	0.77 % ($\approx 6.5\ K$)	0.26 % ($\approx 2.5\ K$)	0.85 % ($\approx 6.9\ K$)	0.39 % ($\approx 3.3\ K$)
2	Peng-Robinson (PR) EOS	0.54 % ($\approx 4.5\ K$)	0.01 % ($\approx 0.1\ K$)	0.61 % ($\approx 5.1\ K$)	0.02 % ($\approx 0.2\ K$)
3	Virial EOS	0.68 % ($\approx 5.8\ K$)	0.11 % ($\approx 1.0\ K$)	0.74 % ($\approx 6.3\ K$)	0.18 % ($\approx 1.5\ K$)

Fig. 4.14 shows the comparison between model predictions versus measurements truth with respect to both METCNN and NN without mass-energy-thermodynamics constraints (abbreviated as METC for the rest of the report) models for the simulation data of the reactor outlet temperature (T_{out}).

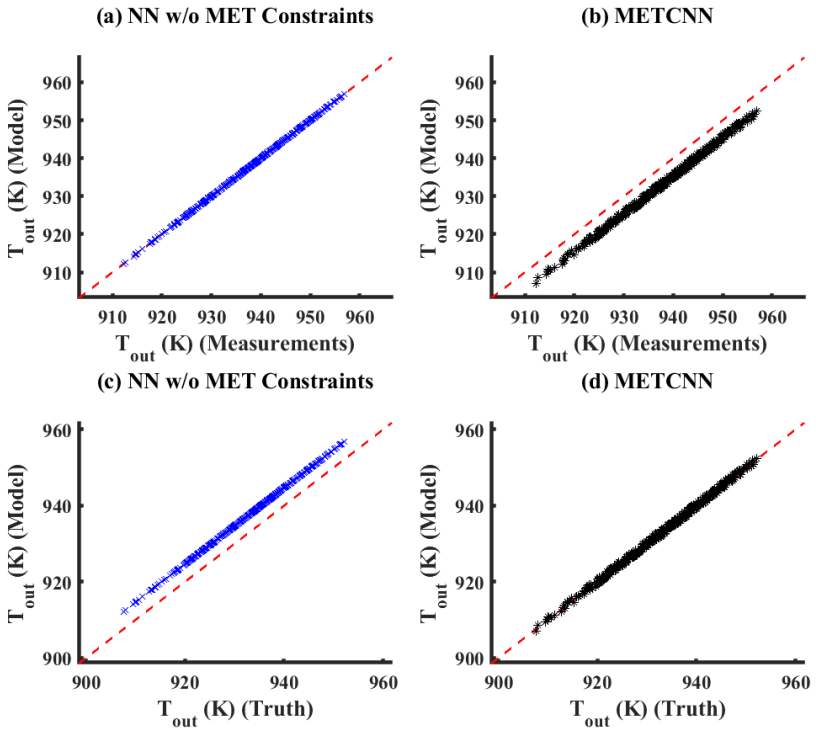


Fig. 4.14. Comparison of results between METCNN and NN w/o mass-energy-thermodynamics constraints (METC) for simulation data of reactor outlet temperature (T_{out})

Similar parity plots for the outlet concentration of A (C_A) in the product stream are included under **Fig. C. 6** in **Appendix C**. As before, the unconstrained NN model shows an excellent match with the biased measurements, thus leading to significantly higher errors with respect to the system truth represented by an approximate RMSE of 0.6% (i.e., around 5 K). On the other hand, the METCNN model accurately captures the system truth, even though trained against biased / noisy measurements. Additionally, Fig. 4.15 shows the comparisons between the APEs in C, H, O, and energy (Q) balances obtained from the unconstrained NN, MECNN, and METCNN models for simulation data. From Fig. 4.15, it is evident that both the MECNN and METCNN models exactly satisfy the mass (atom) balance constraints during simulation, while the unconstrained NN model shows APE values with respect to C, H, and O balances as high as around 17% for each of them. However, Fig. 4.15 (d) demonstrates that the MECNN model involving the enthalpy conservation constraints expressed in terms of ideal gas enthalpy equations without incorporating appropriate additional thermodynamic models to account for the enthalpy departure based on a suitable EOS cannot exactly conserve the overall energy balance for the system, as represented by APE values in energy balance as high as 5.3 %. Therefore, although the MECNN model shows relatively improved predictive performance as compared to the unconstrained NN (which records maximum APE in energy balance for simulation data as around 10.7 %) in terms of conserving system physics (especially, enthalpy for this case), the optimal MECNN model still cannot guarantee the exact conservation of the overall energy of the system under consideration. On the contrary, the METCNN model exactly conserves the enthalpy of the system due to consideration of the respective mass, energy, and thermodynamics constraints. Similar results for the training data comparing the APEs obtained with respect to the

METCNN, MECNN, and NN model without METC have been presented under **Fig. C. 7** in**Appendix C**.

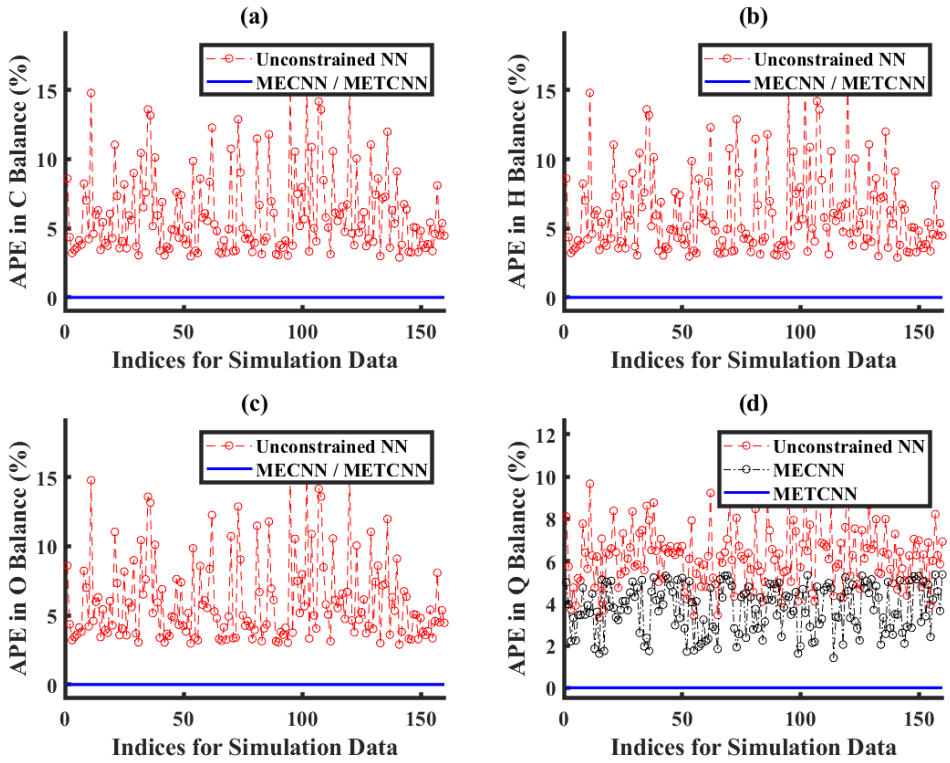


Fig. 4.15 . Comparison between METCNN, MECNN and NN w/o METC in terms of violating (a) carbon (C), (b) hydrogen (H), (c) oxygen (O) and (d) energy (Q) balance constraints during simulation

Specifically, it is to be noted that there are no absolute unique formulations of the thermodynamics model for a system under consideration, but the proposed approaches can result in a decision of the best model subject to a given training dataset. However, irrespective of the specific thermodynamics constraints imposed during parameter estimation (training problem) as well as model simulation (forward problem), the optimal METCNN models will always ensure that the model predictions are thermodynamically consistent, albeit no guarantee can be provided in terms of exactly satisfying the system truth. While the choice of the specific set of candidate thermodynamics models to be included in the integer programming problem during the development of METCNN models depends on the user, this work provides a framework of the different candidate constraints that can be imposed followed by the discovery of the best thermodynamic model, as well as approaches for incorporating such constraints while formulating training and forward problems of suitable data-driven models, especially for interconnected systems.

4.3.5 Results and Discussions: Dynamic Modeling

The steady-state ECNN / METCNN models have been extended to dynamic modeling of both the case studies under consideration, subject to formulation of the necessary mass, energy, and thermodynamics constraints. However, since both the systems are spatially distributed, it is assumed that the holdup information (measurements) are not available during development of optimal dynamic METCNN / ECNN / MECNN models and therefore, the mass and energy conservation laws can only be applied at steady state. However, there may exist certain specific thermodynamics constraints (such as for the combined HX-flash system), such as VLE equations, etc. that are instantaneously achieved in practical operations and hence, can be imposed as additional constraints at every time step during formulation of the dynamic training and forward problems.

Case Study 1: Adiabatic Superheater / Reheater System

Similar to steady-state modeling of the distributed superheater / reheater system, a time-invariant bias equaling to around 5°C has been added to each output temperature variable to generate biased dynamic training data for the distributed superheater system, along with an additional Gaussian noise distribution with $\mu = 0$ and $\sigma^2 = 0.25$. The principal motivation for developing a distributed model as opposed to a lumped parameter model is to gain insights into predicting the temperature distributions at intermediate discretization grids within the superheater / reheater system. For all results shown in this section, the neural networks without energy constraints are represented by the networks with the architecture as the hybrid ECNN model, which have been trained against the measurement data in the absence of any energy balance constraints. As before, the hybrid parallel ECNN yields accurate results with respect to system truth including the transients even though the presence of the bias in the data is clear, thus leading to around 0.08% and 0.73% RMSE respectively with true data and measured data. Reverse trend is observed for the NN model without energy constraints which record 0.74% and 0.12% RMSE respectively when compared to the true data and measurements respectively. Furthermore, the distributed model also ensures that the energy conservation laws are exactly satisfied not only at the system boundaries, but also at all intermediate discretization grids under consideration.

Since this case study considers a spatially distributed system, the holdup information cannot be derived from system boundary conditions and hence, no guarantee can be provided on exactly conserving the overall mass of the system during transients. The distributed ECNN model for the superheater / reheater system also provides the temporal temperature distribution in the spatial coordinates within the system during both training and forward (simulation) problems, as shown in Fig. 4.16. As expected, with respect to the schematic of the distributed model given in Fig. 4.10 (b), the outlet steam temperature at $(1,1)$ is consistently higher than that at $(2,1)$ even though $T_{St,in}$ for both these grids is the same, since the $(1,1)$ grid / block has the highest heat transfer gradient when the flue gas first comes in contact with steam and the gradient gradually decreases at subsequent discretized element(s) in the horizontal direction. Accordingly, the same trend is also observed for $T_{St,out}$ at $(1,2)$ and $(2,2)$ respectively. Other training / simulation results at different locations in the spatial coordinates for both hybrid series and parallel ECNNs for this type of error characterization have not been discussed in this document for brevity.

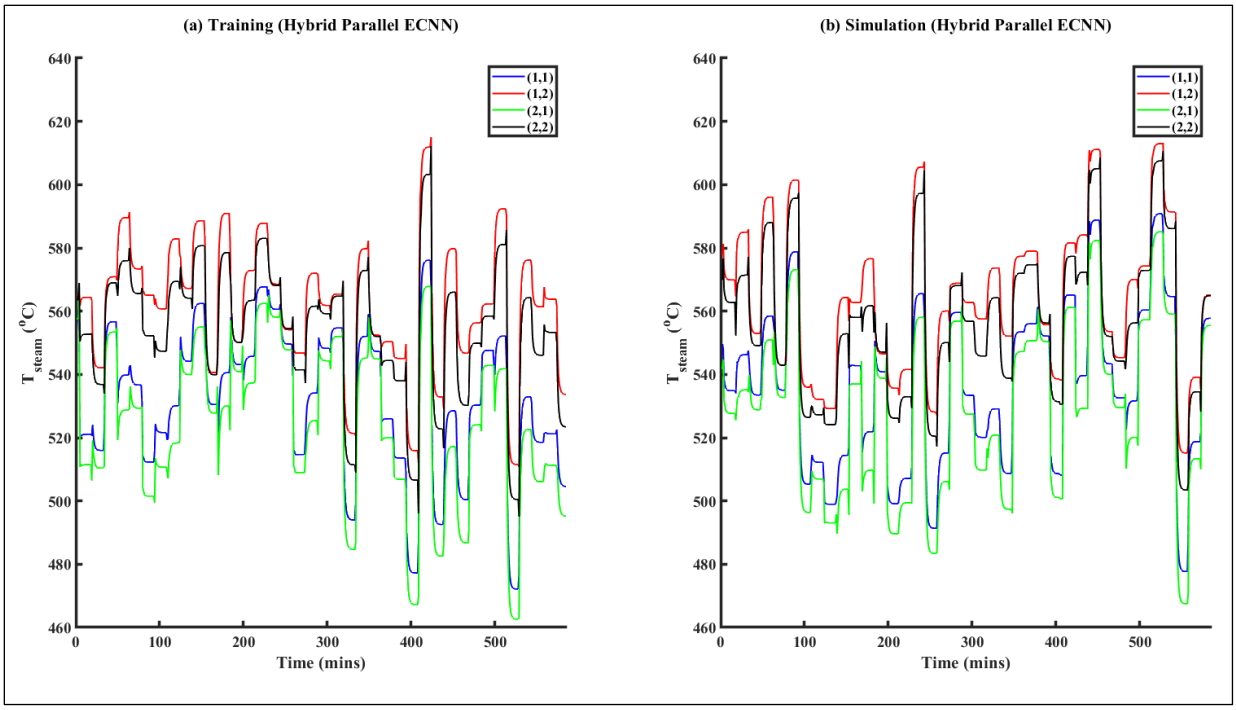


Fig. 4.16. Results from dynamic (hybrid parallel) ECNN for $T_{St,out}$ distribution in spatial coordinates during (a) training and (b) simulation

Case Study 2: Interconnected System for Production of Acetic Anhydride

For the second case study, i.e., modeling the tubular reactor system, the energy (enthalpy) balance constraints have been updated to include the additional appropriate thermodynamics-based enthalpy departure functions given a specific EOS (i.e., PR EOS) during formulation of physics-constraints for METCNNs, as compared to the consideration of only ideal gas enthalpy balance in case of MECNNs. Therefore, in the absence of holdup measurements, the corresponding mass-energy-thermodynamics constraints can only be applied at steady-state, even while developing the dynamic model for the tubular reactor system. The unavailability of system holdup information does not necessarily imply that the proposed structures and algorithms cannot be applied for modeling such systems, however, in such cases most physics conservation constraints can only be applied during steady-state conditions. It is worth mentioning that in presence of system holdup information and/or in cases when certain thermodynamic constraints need to be exactly satisfied at all time steps, the corresponding algorithms can be appropriately modified as discussed in Fig. 4.9 (b) and Eqs. (4.19a)- (4.19e) in Section 4.3.2. Fig. 4.17 presents a summary of simulation results.

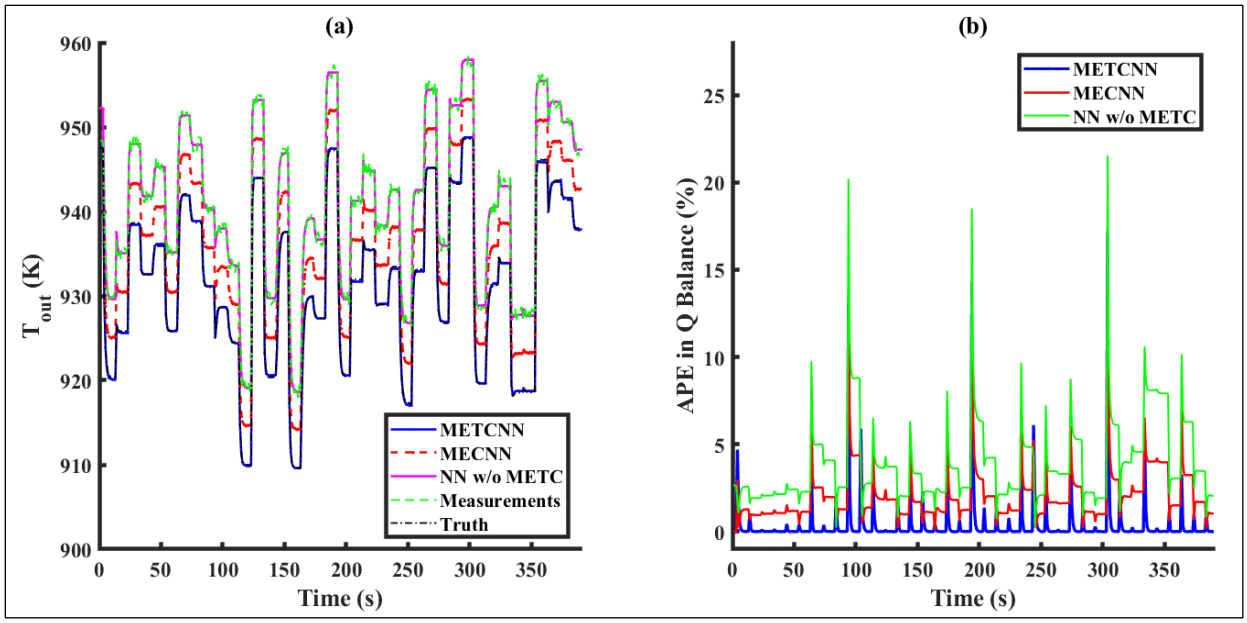


Fig. 4.17. Comparison of results between dynamic METCNN, dynamic MECNN, and hybrid series NN w/o METC (a) for simulation data of T_{out} and (b) in terms of violating energy (Q) balance constraints during simulation

It is to be noted that for comparison of model performance in this case, all three candidate network models, i.e., METCNN, MECNN, and NN without METC, correspond to the same architectures for the neural network models involved in the hybrid series (FFNN followed by RNN) architecture. From Fig. 4.17 (a), it can be clearly observed that the results from the unconstrained NN model match very well with the transient biased measurements but leads to significantly higher mismatch with the true data (represented by an approximate RMSE of 0.5 %, or about 4 K). On the contrary, the METCNN model predictions show an excellent match with the system truth leading to around 0.01 % RMSE or about 0.2 K, even though trained against the biased measurements. While consideration of both candidate thermodynamic models other than that using PR EOS represented by appropriate thermodynamic constraints derived from the ideal gas equations (i.e., the dynamic MECNN model) as well as virial EOS show relatively superior performance as compared to the unconstrained NN model, the presence of inaccurate thermodynamics constraints result in significantly higher mismatches with the true data as compared to the optimal METCNN, thus leading to about 2.4 K and 1.6 K RMSE respectively, for simulation data.

Similar relative performance between these three models can also be seen in Fig. 4.17 (b). As expected, the METCNN exactly satisfies the energy (enthalpy) balance constraints at steady state, while still recording some peaks in APE for energy balance during the transients due to lack of holdup information. On the contrary, both MECNN and NN without METC show consistent errors with respect to energy conservation constraints, even at steady states, leading to an overall maximum APE in energy balance of 11 % and 22 %, respectively. Additionally, it has also been observed that the METCNN model considering the virial EOS as the candidate thermodynamics model also resulted in significant violations of the system physics at both steady-state and dynamic zones, represented by maximum APE values in energy balance as high as 3.1 % for steady-state and 8.9 % for transient data, respectively. In addition to mass balance constraints, the development of optimal METCNN models in presence of appropriate enthalpy balance equations including departure models derived from a suitable thermodynamic EOS

ensures significantly accurate model performance during both training and simulation. However, similar to steady-state modeling, it is to be noted that the specific choice of thermodynamics constraints to be incorporated may vary depending on the user, but the proposed approaches provide a generic framework for ensuring that such constraints are exactly satisfied during both training and forward problems. More details on the algorithms and results obtained while exactly satisfying physics constraints in presence of system holdup measurements can be found in our recent publications[168–170].

4.3.6 Conclusions

In this study, we have developed algorithms for both training and forward problems of steady-state and dynamic METCNN / ECNN models capable of exactly conserving the mass, energy, and thermodynamics constraints in interconnected systems, even though the corresponding training data consists of various uncertainties and violate the physics conservation laws. In addition, we have also demonstrated that the proposed approaches can accommodate an outer layer integer programming problem for selecting the best thermodynamics model from a family of candidate models that can be posed as additional constraint equations during solving the equality constrained optimization problems. Both the steady-state and dynamic results observed from predictions of optimal METCNN / ECNN models for both the case studies considered in this study show that the exact satisfaction of mass and energy conservation constraints does not necessarily ensure that the model outputs are thermodynamically consistent.

For the tubular PFR system, the optimal METCNN model considering the Peng Robinson EOS as the best thermodynamic candidate model shows an excellent match with the system truth, thus leading to a corresponding RMSE value of around $0.1\ K$. On the contrary, the unconstrained NN model leads to significantly higher mismatch with true data represented by a corresponding overall RMSE of around $5\ K$. Additionally, the optimal METCNN models also exactly satisfy the physics (mass, energy, and thermodynamics) constraints at steady state, while the APE in energy balances calculated from the model predictions of the unconstrained NN as well as MECNN (i.e., considering only mass and energy balance without any additional thermodynamics constraints) range to as high as $10.7\ \%$ and $5.3\ \%$ respectively. Similar results have been observed for dynamic modeling implementations using METCNN models. For the reactor system, while the unconstrained NN, MECNN and METCNN with thermodynamics constraints derived from virial EOS resulted in significantly higher (maximum being 22%, 11%, 8.9% respectively) APE in energy balance for both steady-state and dynamic data, the optimal METCNN model consist of the appropriate thermodynamics constraints formulated from the PR EOS (found to be the best thermodynamic model for the system under consideration) exactly conserved the energy balance at steady-state, along with some mismatches at transients due to unavailability of holdup information.

It should be noted that the thermodynamics constraints (such as those for enthalpy model, K-value, etc.) imposed by considering specific EOS, departure function models, etc. will always represent the system. The choice of the thermodynamic constraints may have been established by the user based on literature information or their own study. Therefore, even in cases where the proposed constraint equations do not represent the system truth, the algorithms developed in this work will still ensure the exact satisfaction of system thermodynamics based on the current understanding of the user (i.e., true in the sense of user belief / understanding). Furthermore, the proposed algorithms can easily accommodate a combination of two or more systems (such as the combined HX-flash process considered in this work) and model them as a single system in absence of intermediate measurements. Additionally, it must also be noted that although the corresponding results thus obtained from the optimal models are thermodynamically consistent, no guarantee can be provided with respect to accurately converging to the system truth during both training and forward problems. The only guarantee that can be provided for the optimal METCNN models

developed for any chemical process system is that the model predictions will follow the mass and energy conservation constraints and will also be thermodynamically consistent during both training and forward (simulation) problems.

4.3.7 Code Availability Statement

All data used in this work are shown in the form of plots and tables. The generic framework for imposing physics constraints during training and simulation of neural network models is publicly available in various GitHub repositories.

The Python and MATLAB codes for steady-state and dynamic mass constrained neural networks (MCNNs) can be found on GitHub at: https://github.com/mukherjeeangan26/Steady_Dynamic_MCNN.

Similar codes for steady-state and dynamic mass-energy constrained neural networks (MECNNs / ECNNs) can be found on GitHub at: https://github.com/mukherjeeangan26/Steady_Dynamic_ECNN.

Since the proposed thermodynamics constraints may be specific depending on the type of system (for example, VLE constraints can only be applied for two-phase mixtures, etc.), the MCNN and MECNN codes can readily be extended to accommodate appropriate additional thermodynamics constraints or update current formulations of constraint equations with due consideration of system thermodynamics.

Additionally, the MATLAB codes developed for training the unconstrained hybrid series and parallel all-nonlinear static-dynamic neural network models are also available on GitHub at: https://github.com/mukherjeeangan26/Hybrid_StaticDynamic_NN_Models.

4.4 Gaussian Radial Basis Neural Network (GRAB-NN) Models

Although multilayered feedforward ANNs act as universal approximators for many classification problems[272], existing literature has shown that Gaussian radial basis functions (RBFs) have outperformed conventional ANNs for function approximation problems in terms of robustness, fault tolerance, and computational expense for noisy time series data using finite number of hidden units[273]. Therefore, for modeling uncertain systems, the performance and efficiency of the hybrid series and parallel all-nonlinear networks can be further improved by considering probabilistic / stochastic network structures. This chapter discusses the development of novel architectures and training algorithms for hybrid Gaussian radial basis neural networks. The primary motivation behind the development of such hybrid models combining ANN (deterministic) and RBF (stochastic) activation functions is to achieve the trade-off between local (characteristic of typical RBF models) and global (characteristic of ANN models) approximations, especially for nonlinear dynamic noisy chemical processes.

4.4.1 Literature Review

Existing literature shows that although multilayered feedforward neural networks have performed relatively better in several classification problems as universal approximators[272,274], RBFs have outperformed conventional ANNs for function approximation problems in terms of robustness, fault tolerance, and computational time involved while dealing with noisy input data set[273,275–277]. RBFs have also been shown to produce satisfactory results while being trained by using noisy time series data[278]. However, the performance of RBFs can become highly sensitive to the number of parameters

estimated during optimal model synthesis[276,277] as well as the placement of RBF centers[279]. If the number of RBF centers is large, it would lead to large number of model parameters (weights and biases) resulting in potential overfitting during training and can lead to inferior predictive capabilities[192,280], especially in presence of uncertainties in training data.

Although there exist different choices of kernels for RBFs, the Gaussian function is the most widely used kernel for most RBF applications in function approximations [281]. For the classical RBF architectures, the number of weights and bias parameters estimated during model training depends on the number of RBFs (neurons) in the hidden layer, each identified by a specific coordinate of center and width [281]. The development of standalone optimal RBF networks mainly consists of two steps. First, the number of RBF centers is determined along with estimating the coordinates of the centers and widths of RBFs in the hidden layer. This, in turn, is followed by the second step involving the estimation of optimal connection weights for the output layer of the network model. The predictive performance of the RBF networks critically relies on the number of RBF centers and the corresponding coordinates of the centers and widths [282]. The most commonly used algorithms implemented in open literature for selecting the RBF centers and widths involve both soft clustering approaches such as fuzzy clustering [283–288], particle swarm optimization (PSO) [284,289–291], etc. as well as hard clustering techniques such as k-means [292–294] or k-nearest neighbor based approaches [295–299], along with several other metaheuristic algorithms like firefly [300] and prey-predator [301] algorithms. Such clustering / metaheuristic approaches may suffer from the curse of dimensionality for modeling higher order systems with larger input space and can be highly sensitive to noisy data during optimal model synthesis and parameter estimation.

This work develops a recursive training algorithm that downselects the number of RBF centers based on penalizing overfitting and parametric uncertainties [214], thus leading to the development of sparse data-driven models. An efficient sequential mixed integer nonlinear programming (MINLP) algorithm, referred to as the ‘rigorous approach’ in the rest of the chapter, using bidirectional branch and bound (BB&B) approach has been developed to determine the optimal number of RBF centers. Two types of distributions are considered for selecting the coordinates of RBF centers (and widths) – deterministic and stochastic. While the deterministic distribution refers to assigning constant values to the centers / widths of the RBFs, the stochastic distribution considers sampling the centers from Gaussian distributions having the same mean and variances as the model inputs. The solution space is parameterized by representing the RBF centers / widths and the means / variances of the center distributions as additional model parameters to be estimated during training of the optimal RBF network models with deterministic and stochastic centers, respectively, in addition to the connection weights in the output layer of the network models. Nevertheless, such problem formulation demands the optimal estimation of the deterministic centers and widths of the Gaussian RBF network models, along with the weights for the output layer to be monolithically estimated using the interior point optimization algorithm, thus leading to the sparse model structure. On the other hand, for the Gaussian RBF models with stochastic centers, the means and variances of the respective center distributions are updated using a modified Adam stochastic gradient descent (SGD) with mini-batch algorithm.

However, for significantly large-sized data and / or deeper network structures (i.e., higher number of model parameters), the proposed rigorous MINLP approaches aimed at developing the optimal parsimonious model structure may incur excessive computational expense. In such cases, an ‘approximate approach’ has been proposed where the RBF centers / widths are either assigned some fixed deterministic / probabilistic values (without optimization), followed by only estimating the connection weights in the output layer of the network model. The number of centers is then sequentially increased to obtain the

specific model architecture that minimizes the corrected Akaike Information Criteria (AICc) [214,233]. It is to be noted that although the approximate approach ensures faster computation since the coordinates of RBF centers / widths need not be estimated through optimization, such approaches may not necessarily lead to the parsimonious model architecture during parameter estimation. Among the different approaches discussed in the existing literature for the calculation of the optimal weights for the RBFs [281,302,303], the orthogonal least squares (OLS) approach that follows forward stepwise regression for minimizing the sum squared error during model training is one of the fastest approach. The OLS algorithm also enables the estimation of optimal connection weights for the RBF network models to be performed in a single iteration by considering the entire set of training data. However, there is an abundance of proposals developed in the last few decades for optimal estimation of model parameters during training of RBF models based on their respective merits and demerits, which can be found in numerous references elsewhere [281,284,304–308].

It may be significantly challenging to adequately represent many nonlinear dynamic process systems using only a specific type of simple (i.e., consisting of only one or very few hidden layers) feedforward or feedback data-driven model [190]. Furthermore, for function approximation of nonlinear dynamic systems, hybrid models that synergistically combine different types of data-driven sub-models have been reported to perform significantly superior compared to their respective standalone counterparts [166,210,309,310]. However, most of such hybrid network architectures involve connecting two separate and different models (such as RBF, ANN, PSO, autoregressive networks, etc.) in serial or parallel configurations [204,311–313], thus improving overall predictive capabilities but at the cost of significantly larger number of additional parameters as compared to a standalone simple model. While only a few works exist in the literature which discusses the development of deep neural network (DNN) model architectures with mixed activation function[314] in different layers, no paper has so far considered hybridization of RBF and ANN activation functions within the same layer in a network architecture or the corresponding training algorithm for such hybrid models. In this work, the proposed architectures and training algorithms for standalone Gaussian RBF networks are coupled with conventional feedforward ANN models, thus leading to the development of hybrid Gaussian radial basis neural network structure, abbreviated as GRAB-NN for the rest of this chapter. However, instead of combining separate standalone RBF and ANN models arranged in series or parallel to each other for generating the hybrid model architecture, the proposed GRAB-NN models consists of a single hidden layer containing a combination of activation functions in the hidden nodes such as the Gaussian function (representing the RBF nodes), as well as some of the most widely used standard ANN activation functions such as logistic sigmoid and hyperbolic tangent functions (representing the ANN nodes). It is to be noted that the proposed architectures are generic and sufficiently flexible to incorporate different combinations of additional activation functions as well as other variants of hybrid data-driven model structures within the same hidden layer(s) during development of optimal GRAB-NN models. Furthermore, although typical RBF network models have been implemented for various applications such as dynamic process modeling and controls[315–318], the hybrid models especially those involving one or more types of RBF networks as well as the DNN models with mixed activation functions mentioned above have mostly been applied on steady-state data. All model structures and algorithms presented in this chapter show sufficient flexibility and accuracy while modeling complex nonlinear transient chemical process systems.

The data typically obtained from lab-scale experiments or commercial plant measurements may not necessarily conserve system physics. If physics constraints are not exactly satisfied during machine learning (i.e., parameter estimation), predictions obtained from the optimal data-driven models may violate the physics conservation laws of the system and hence, not meaningful[167,239]. Therefore, the

training approaches are developed in this work to accommodate physics constraints on model outputs during estimation of optimal parameters (weights and biases) of both standalone RBF networks as well as hybrid GRAB-NN models. For example, while developing accurate data-driven models for any chemical process systems, the mass conservation constraints can be expressed as linear elemental atom balance equations in terms of the input and output boundary conditions and incorporated during training of optimal models. One of our previous publications[168] have presented development of mass constrained neural networks (MCNN) for exact satisfaction of mass balance constraints while solving both inverse and forward problems for steady-state and dynamic chemical systems. In this work, the mass balance constraints are imposed on the standalone RBF and hybrid GRAB-NN models such that the overall outputs exactly satisfy the mass conservation laws during training using the noisy transient data that violate mass balance as well as during validation. It is worth mentioning that the proposed model architectures are sufficiently flexible to incorporate other linear and / or nonlinear constraints during parameter estimation, as necessary. Additionally, it may be significantly challenging to develop accurate physics-constrained dynamic data-driven models aimed at exactly satisfying the constraints imposed, because most conservation laws can only adequately be implemented to steady-state data if sufficient / necessary local measurements in distributed systems are unavailable. This work also focuses on the development of mass constrained dynamic RBF / GRAB-NN models while considering training data corrupted with uncertainties.

Development of efficient training algorithms for the proposed network structures may be considerably challenging. Typical training approaches for data-driven network models maybe categorized into first-order and second-order methods[215,216]. While the first-order approaches such as gradient descent are primarily beneficial for training large sized network models involving millions of parameters, they may require significant tuning of hyper parameters and can lead to poor convergence. On the contrary, second order parameter estimation approaches may provide flexibility for incorporating constraints as well as addressing some of the limitations of first-order methods, but they can be associated with a significantly higher computational cost. For the proposed structures and the underlying training problem, training of mass constrained RBF and GRAB-NN models by solving an equality constrained optimization problem coupled with the solution of the MINLP problem for parameter estimation by using a monolithic approach is challenging and computationally expensive. In addition, especially for the hybrid GRAB-NN models, the most efficient algorithm for estimating the optimal RBF parameters may be different from the most efficient algorithm for estimating the ANN parameters. Therefore, sequential decomposition-based training algorithms are developed in this work that exploit the network architectures and provide flexibility for parameter estimation in the different sublayers (i.e., parameters associated with RBF nodes, ANN nodes, and connection weights in the output layer) independently, while solving an outer layer optimization to ensure convergence of the overall network model. Furthermore, the optimal weights in the output layer of the proposed network models are estimated through a single step constrained / orthogonal least squares (CLS / OLS) algorithm, depending on the presence or absence of linear constraints respectively, during model development by using the proposed sequential approaches. It is to be noted that the proposed decomposition-based training algorithms, referred to as ‘sequential algorithms’ in this chapter, can be applied irrespective of whether the imposed constraints are linear or nonlinear. However, the single-step CLS approach for estimating the output layer weights can only be applied if the constraints are linear. For nonlinear constraints, the same parameters (weights) need to be estimated by solving an equality constrained optimization problem since the CLS algorithm conventionally cannot handle nonlinear constraints during formulation. Performances of the sequential algorithms have been compared with the monolithic ‘simultaneous approaches’ for parameter estimation that involves evaluation of all model parameters (including RBF centers / widths as well as weights / biases) simultaneously. The

proposed data-driven structures and algorithms are applied to modeling various nonlinear transient chemical process systems, including the widely used example problem represented by the isothermal Van de Vusse reactor, the tubular reactor for vapor-phase cracking of acetone, as well as transient variable-load industrial steam superheater systems[147] at partner combined cycle and coal-fired power plant using plant measurement data for model training.

4.4.2 Candidate Network Architectures

This section provides brief descriptions of the two types of data-driven models considered in this work, namely the standalone Gaussian RBF type networks and the hybrid GRAB-NN models. Although various types of activation functions and / or network structures may be combined within the hybrid GRAB-NN models, only the ones that have been used in this work for process modeling are discussed here.

Gaussian RBF Networks

The conventional architecture of RBF network models typically achieves approximations of nonlinear functions in two stages. The first stage involves performing a nonlinear mapping from the input layer to the hidden layer, followed by the second stage which yields a linear transformation of the outputs from the hidden layer to the output layer through the connection weights (parameters). Therefore, learning is only performed at the output layer for conventional RBF networks. Furthermore, unlike classical feedforward multi-layered neural networks that consider numerous input features (variables) directly, RBF networks define sets of 'centers' that are subsequently estimated as independent variables through calculation of the corresponding distances from the model inputs, thus leading to the radial basis function approximation[215,281]. The hidden layer includes a number of RBF units (n_h) with an appropriate kernel / activation function. One of the most used RBF kernels is the Gaussian function, represented as:

$$\varphi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{4.38}$$

where, σ^2 denotes the variance of the Gaussian function under consideration. The Gaussian activation function for i^{th} node in the hidden layer is determined by a center (c_i) and a corresponding width (σ_i). The RBF network classifier compares the Euclidean distance between the input vector $\left(\boldsymbol{x}(\boldsymbol{t})\right)$ and the center distribution of the Gaussian function $(\boldsymbol{c})$ under consideration of the following nonlinear transformation in the hidden layer of the resultant network architecture:

$$\boldsymbol{D} = \begin{bmatrix} ||x_1 - c_1|| & \cdots & \left||x_{n_i} - c_1|\right| \\ \vdots & \ddots & \vdots \\ \left||x_1 - c_{n_h}|\right| & \cdots & \left||x_{n_i} - c_{n_h}|\right| \end{bmatrix} \tag{4.39a}$$

$$\boldsymbol{Dh}(\boldsymbol{t}) = \varphi(\boldsymbol{D}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\boldsymbol{D} \otimes \boldsymbol{D}}{2\sigma^2}\right) \tag{4.39b}$$

where, n_i denotes the total number of nodes / neurons in the input layer, $\boldsymbol{h(t)}$ represents the outputs of the RBF nodes / neurons in the hidden layer, and $\boldsymbol{D \otimes D}$ stands for the elementwise product of the $\boldsymbol{D}$ matrix. The linear mapping in the output layer can subsequently be expressed in terms of the connection weight matrix (w) as given by:

$$\boldsymbol{y_k(t)} = \sum_{j=1}^{n_h} w_{kj} * \boldsymbol{h_j(t)} \tag{4.40}$$

where, $\boldsymbol{y_k(t)}$ refers to the final outputs from the k^{th} node of the output layer of the RBF network model. More details about the conventional architectures and training approaches for Gaussian RBF network models can be found in numerous references[215,281,299,306,308].

Hybrid GRAB-NN Model

Fig. 4.18 shows the generic architecture of the single hidden-layered hybrid GRAB-NN model proposed in this work, consisting of $n_{h_{ANN}}$ ANN nodes (represented by blue circles) and $n_{h_{RBF}}$ RBF nodes (represented by orange circles) in the hidden layer. The number of outputs considered for the GRAB-NN model is represented as n_o.

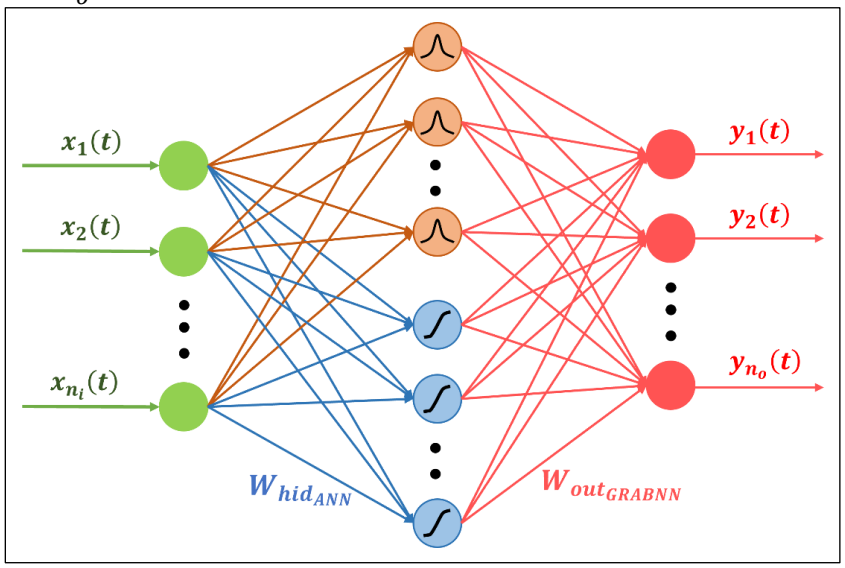


Fig. 4.18. General architecture of the hybrid GRAB-NN model

It is to be noted that the proposed model structure is neither dependent on the type of activation function considered in the hidden nodes nor the specific type of data-driven models that can be incorporated in the hybrid framework. While the sigmoid activation function is considered for all ANN nodes in the hidden layer, the outputs from the RBF nodes $\left(\boldsymbol{h_{RBF}(t)}\right)$ are obtained using the Gaussian kernel as given by Eq. (4.39a). The outputs from the ANN nodes, represented by $\boldsymbol{h_{ANN}(t)}$, may be mathematically expressed in terms of the model inputs $\left(\boldsymbol{x(t)}\right)$ and the connection weight matrix $\left(W_{hid_{ANN}}\right)$ between the input and hidden layers, given by:

$$\boldsymbol{h_{ANN}(t)} = \frac{1}{1 + \exp\left(-W_{hid_{ANN}} * \boldsymbol{x(t)}\right)} \tag{4.41}$$

The outputs from both the RBF and ANN nodes are augmented to generate the overall outputs $(\boldsymbol{h}(t))$ from the hidden layer of the proposed GRAB-NN models, given as:

$$\boldsymbol{h}(t) = [\boldsymbol{h_{RBF}}(t) \quad \boldsymbol{h_{ANN}}(t)]^T \tag{4.42}$$

It is worth mentioning that since the proposed architecture of hybrid GRAB-NN considers a fully connected neural network model, the network structure remains consistent irrespective of the relative positions of the RBF and ANN nodes in the hidden layer. The final output vector $(\boldsymbol{y_{GRABNN}}(t))$ can be subsequently obtained from the proposed GRAB-NN models through a linear mapping of $\boldsymbol{h}(t)$ in terms of the weight matrix $(W_{out_{GRABNN}})$ mapping the hidden layer to the output layer, given by:

$$\boldsymbol{hy_{GRABNN}}(t) = W_{out_{GRABNN}} * \boldsymbol{h}(t) \tag{4.43}$$

4.4.3 Training Algorithms

This section includes brief descriptions of the different training algorithms developed in this work for optimal development of both unconstrained and constrained RBF / GRAB-NN models for modeling complex nonlinear dynamic noisy chemical systems. For all approaches proposed in this work, the optimal model parameters are estimated during the inverse problem (training) by minimizing the AICc[214], as shown in Eq. (4.44). The AICc incorporates an additional bias correction term, as compared to the conventional AIC expression[319], to account for small sample sizes assuming that the estimated parameters are normally distributed around the true parameter values[227].

$$AIC_c = -2\ln\left(L\left(\hat{\theta}\right)\right) + 2q + \frac{2q\ (q+1)}{n-q-1} \tag{4.44}$$

where, $\left(L\left(\hat{\theta}\right)\right)$ represents the log-likelihood estimate evaluated at the maximum likelihood parameters $\left(\hat{\theta}\right)$, q denotes the total number of model parameters, and n refers to the total sample size. While the first term in the definition of AICc (Eq. (4.44)) rewards goodness-of-fit of the model with respect to the training data, the following two terms penalizes model complexity, thus preventing potential overfitting during optimal estimation of model parameters.

It is our observation that currently most of the commercial deep learning libraries / software packages such as the Deep Learning Toolbox in MATLAB (R2021a), PyTorch (v2.2.2), TensorFlow (v2.14.0), or NeuroMANCER[257] (v1.4.2) do not offer sufficient flexibilities for training RBF models which involves simultaneous parameterization of the center / width parameters. Therefore, the training algorithms developed in this work are coded in Python using the algebraic modeling language Pyomo[259] within the open-source integrated platform developed by the Institute for Design of Advanced Energy Systems[320] (IDAES) for parameter estimation. While the deterministic distribution of RBF centers have been estimated using the Interior Point OPTimization[258] (IPOPT) solver in Pyomo, the learning of the stochastic center distributions is accomplished by using the modified Adam SGD[321] with mini-batch training approach. The codes developed for all different training algorithms proposed in this work are

available on GitHub (link provided at the end of the report).

Sequential MINLP for Training of Unconstrained RBF Models

The training of an RBF network model involves the optimal estimation of four types of parameters, namely, the number of neurons in the hidden layer (i.e., the number of RBF centers), the coordinates and widths of the RBF centers, and the connection weights in the output layer of the network model. One straightforward approach for estimation of all model parameters while training the RBF network is to follow a monolithic MINLP approach for solving both the integer variable (i.e., number of RBF nodes in the hidden layer) and real variables (i.e., coordinates of centers / widths and weights) simultaneously. However, conventional MINLP approaches, such as branch-and-bound or branch-and-reduce, may lead to excessive computational expense especially while modeling large-scale complex dynamic systems. In this work, sequential parameter estimation algorithms are proposed for optimal training of RBF network models in which the number of hidden-layer nodes (n_h) are down-selected based on a recursive approach for minimizing the objective function. In addition, for all RBF network models considered in this work, the width of each RBF node (σ) is assumed to be the same.

Two different approaches are proposed, namely the 'rigorous' approach and the 'approximate' approach. The rigorous approach deploys a bidirectional branch and bound algorithm for simultaneous estimation of the RBF centers, widths and weights by solving an optimization problem. The proposed algorithm largely reduces the number of evaluations of the objective function corresponding to the variations in n_h, as compared to evaluation of the objective for every value of n_h. However, the rigorous approach may be computationally inefficient for large sized datasets, as well as for network models consisting of more than one hidden layer. For such cases, an approximate approach is proposed. This algorithm only considers the optimal estimation of the connection weights in the output layer of the RBF network, subject to assigning the coordinates of centers and widths based on a deterministic / probabilistic distribution. Since the optimal weight parameters can be obtained by leveraging the model structure using least squares regression, the approximate recursive approach leads to much faster computation compared to the rigorous algorithm.

- **Rigorous Approach**

The rigorous approach proposed in this report seeks to develop optimal sparse RBF network models by minimizing the number of neurons in the hidden layer, along with simultaneous optimization of the coordinates for RBF centers, widths, and output weights. However, unlike many conventional MINLP approaches[234,322] that initiates the search start by setting the number(s) of integer variables to their respective lower bounds and gradually increasing the number(s) based on certain objective function, in this work, the BB&B approach is developed that enables the initialization of the training algorithm from an intermediate value of the integer variable (here, n_h) under consideration. The objective function, as given by Eq. (4.44), is split into two terms (referred to as τ_1 and τ_2) as expressed in Eq. (4.45a). The first term, i.e., τ_1 denotes the log-likelihood term, whereas the second term, i.e., τ_2 refers to the parameter complexity terms in the definition of AICc. It is to be noted that τ_2 can be explicitly expressed in terms of n_h (i.e., $\tau_2 = f_2(n_h)$), since the total number of model parameters (q) is a function of n_i, n_h and n_o, and the number of nodes in the input (n_i) and output (n_o) layers of all data-driven network models proposed in this work are exactly equal to the number of model input and output variables, respectively.

$$J = AIC_c = \tau_1 + \tau_2 \tag{4.45a}$$

where,

$$\tau_1 = -2 \ln\left(L(\hat{\theta})\right) \tag{4.45 b}$$

$$\tau_2 = 2k + \frac{2k\,(k+1)}{n-k-1} \tag{4.45 c}$$

However, the first term in the definition of J in Eq. (4.45a), i.e., τ_1 may not explicitly be expressed in terms of n_h. An exact evaluation of τ_1 for every value of n_h will require solving the optimization problem to obtain the model parameters of the RBF model, thus leading to significantly high computational expense. Therefore, to reduce the number of exact evaluations of the objective function, a quadratic approximation is assumed for $\tau_1 = f_1(n_h)$, as given by:

$$\tau_1 = f_1(n_h) \approx \alpha_0 + \alpha_1\left(n_h - n_{h_1}\right) + \alpha_2\left(n_h - n_{h_1}\right)\left(n_h - n_{h_2}\right) \tag{4.46}$$

where, α_0, α_1, and α_2 are parameters of the quadratic model for τ_1 that can be obtained by evaluating τ_1 at n_{h_1} and n_{h_2}. Since analytical derivatives can be obtained for J with respect to n_h under the assumptions provided in Eq. (4.45a) and (4.46), a Newton-based unconstrained nonlinear optimization problem can be solved considering n_h as a continuous variable to obtain the optimal value, say, $n_{h_{opt}}$. Next, the objective function J is evaluated at $\left\lfloor n_{h_{opt}} \right\rfloor$ and $\left\lceil n_{h_{opt}} \right\rceil$ to obtain the specific value of n_h at which an exact evaluation is performed for the objective function, along with simultaneous estimation of model parameters. The algorithm considers function evaluations obtained at preceding iterations for the current step wherever applicable and is continued till some termination criterion is achieved.

During the exact evaluation of the objective function given in Eq. (4.44), two different types of parameter estimation algorithms are considered based on whether the RBF centers follow a deterministic or probabilistic distribution. For a given number of nodes in the hidden layer of the RBF network model (i.e., n_h), the individual coordinates of the centers, widths, and weights are estimated / optimized using interior point approach if the RBF centers follow a deterministic distribution. On the contrary, if the RBF centers follow a stochastic distribution, the means and variances of the center distributions along with the width and connection weights are optimized using the Adam SGD with mini-batch approach. JAX[323,324], the Python-based library for automatic differentiation, has been used for gradient calculations in the proposed SGD algorithm in this work. However, for the latter case, the step length considered for backtracking line search is updated at each iteration following the Armijo-Goldstein condition[325–327], instead of using the default constant value (in the order of 10^{-3}) to ensure faster convergence. It is to be noted that the primary objective of the rigorous algorithm for structural optimization is to reduce the number of evaluations of the objective function based on efficient pruning during the BB&B approach. The detailed flowchart of the 'rigorous algorithm.' is shown in Fig. 4.19.

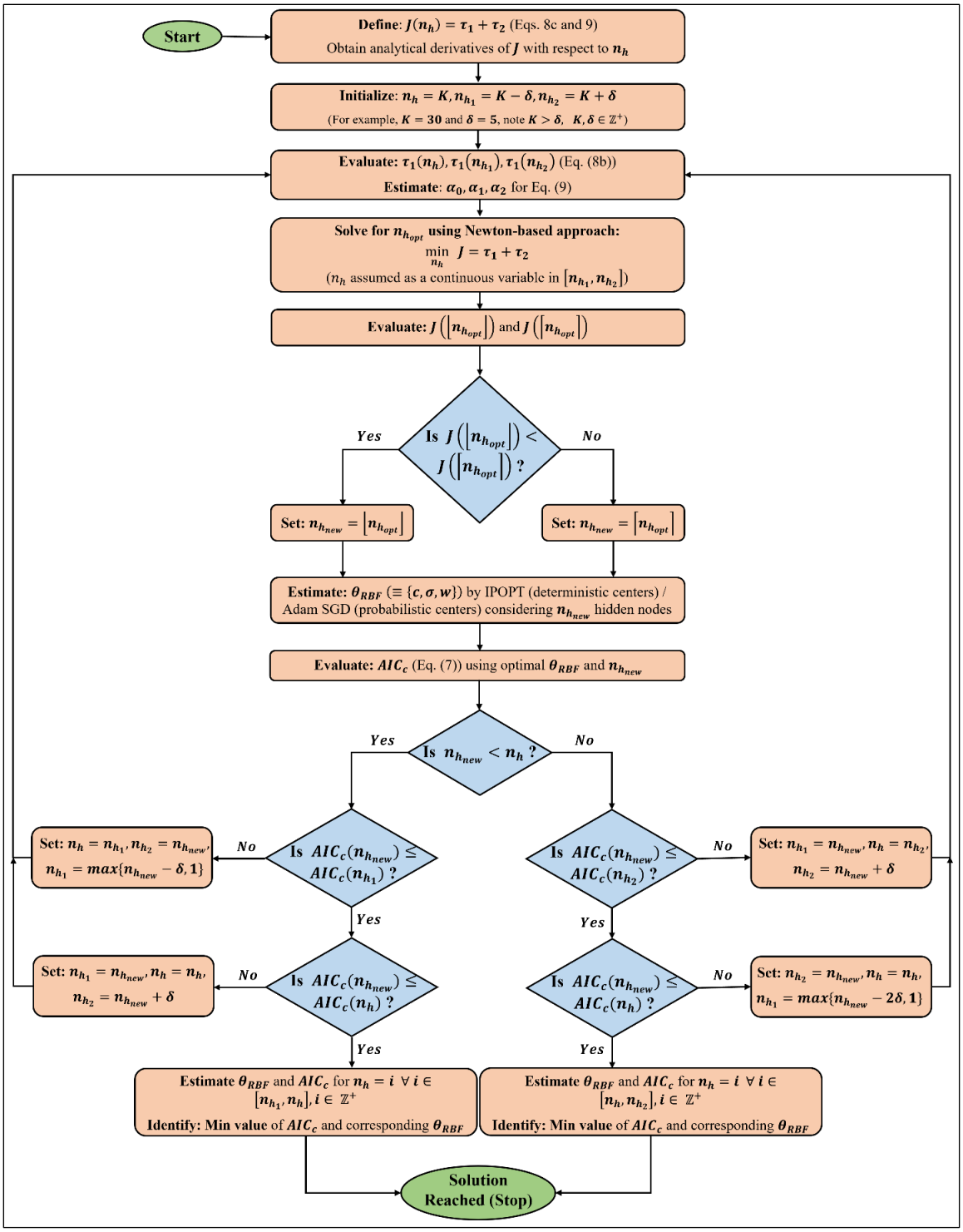


Fig. 4.19. Flowchart describing the rigorous approach for parameter estimation of RBF network

- **Approximate Approach**

The rigorous approach for estimation of optimal parameters of unconstrained RBF models, as discussed above, may become computationally inefficient in the presence of more than one hidden layer in the network architecture and/or for large-sized training data. For such cases, an approximate approach for parameter estimation is developed for training the optimal RBF network models. The primary difference between the rigorous and approximate algorithms is that the former (i.e., rigorous) approach considers the coordinates of RBF centers and widths as additional model parameters to be optimized during parameter estimation (training), while the latter (i.e., approximate) approach does not. In contrast, the 'approximate' algorithm assigns a-priori values to the RBF centers / widths for a given number of hidden nodes (n_h) in the model structure, followed by optimal estimation of the connection weights in the output layer. Furthermore, since the proposed RBF network models consists of a linear activation function in the output layer (as represented in Eq. (4.43)), the model structure can be leveraged in this approach by utilizing the OLS method for estimating optimal weights during model training. Since the OLS approach essentially converges in a single iteration by theory, the approximate approach can perform comprehensive enumeration by varying the size of the hidden layer, without incurring excessive computational expense.

The number of nodes / neurons in the hidden layer (i.e., n_h) has been varied from a minimum value ($n_h = 1$) till a sufficiently large maximum limit (often equal to the size of training data). The optimal number of hidden layer nodes is determined as the one that yields the minimum value of AICc through the comprehensive enumeration approach. For the RBF network models with deterministic centers, the coordinates of RBF centers are assigned constant (fixed) values sampled from a Latin hypercube sampling (LHS) method. However, it must be noted that the proposed algorithms are sufficiently flexible to accommodate any sampling technique for the RBF centers such as various soft and hard clustering approaches[292,295,328,329]. On the contrary, while training the RBF network models with probabilistic centers, the respective coordinates are assigned as n_h samples randomly drawn from Gaussian distributions characterized by the same/similar means and variances as the model input variables. The 'approximate approach' developed in this work for training unconstrained RBF network models is described in Algorithm 1.

Algorithm 1: 'Approximate' approach for training unconstrained RBF network models

Set the maximum number of centers to be considered for the RBF network model $(\boldsymbol{n_{h_{max}}})$
Initialize: $\boldsymbol{n_h = 1}$
while $\boldsymbol{n_h \leq n_{h_{max}}}$:
    Assign coordinates for RBF centers and widths (deterministic / probabilistic)
    Estimate $\boldsymbol{\theta_{RBF}}$, i.e., the connection weights for the output layer by OLS algorithm
    Calculate the value of AIC_c (Eq. (4.44)) and rank them based on the values of n_h
    $n_h = n_h + 1$
end while
Return $\boldsymbol{\theta_{RBF}}$ (optimal parameters) and n_h corresponding to minimum value of AIC_c

Simultaneous vs Sequential Training of Unconstrained Hybrid GRAB-NN Models

As noted before, one straightforward approach for training the hybrid GRAB-NN models is by implementing a monolithic parameter estimation algorithm, where all model parameters (i.e., coordinates of centers / widths for the RBF nodes, hidden layer weights for the ANN nodes, and output layer weights for the overall GRAB-NN) are estimated simultaneously by solving the following optimization problem:

$$\min_{\boldsymbol{c_{RBF}},\boldsymbol{\sigma_{RBF}},\boldsymbol{W_{hid_{ANN}}},\boldsymbol{W_{out_{GRABNN}}}} \boldsymbol{J} = \left(\boldsymbol{y_{tar}}(t) - \boldsymbol{y_{GRABNN}}(t)\right)^T R^{-1}\left(\boldsymbol{y_{tar}}(t) - \boldsymbol{y_{GRABNN}}(t)\right) \quad (4.47a)$$

$$\text{s.}\,t.\ \boldsymbol{y_{GRABNN}}(t) = \boldsymbol{f_{GRABNN}}\left(\boldsymbol{x}, \boldsymbol{c_{RBF}}, \boldsymbol{\sigma_{RBF}}, \boldsymbol{W_{hid_{ANN}}}, \boldsymbol{W_{out_{GRABNN}}}, t\right) \quad (4.47b)$$

where, $\boldsymbol{y_{tar}}(t)$ denotes the target (training) data, R refers to the weightage term that represents measurement uncertainties, c_{RBF} and σ_{RBF} represent the coordinates of the centers and widths, respectively, of the RBF nodes in the hidden layer, $\boldsymbol{x}(t)$ signifies the inputs to the GRAB-NN model, and $\boldsymbol{f_{GRABNN}}(\cdot)$ refers to the overall model equations for the hybrid GRAB-NN model that can be appropriately given by combining Eqs. (4.39a)– (4.43). However, it was observed during model training that the simultaneous parameter estimation approach led to significantly slower convergence while using the IPOPT solver in Pyomo within the IDAES framework. It is to be noted that the simultaneous monolithic approach can also become highly sensitive to the initial guesses for the model parameters which are difficult to obtain during parameter estimation (training) of optimal models. Furthermore, such algorithms may also require significant tuning of model hyperparameters, thus leading to excessive computational expense, especially when the network is large and a large amount of dynamic data are considered for training [215,216,218]. Although existing literature discusses some approaches for training hybrid neural network models where same / different algorithms have been implemented to train separate networks connected in series or parallel to each other [166,233], there is no work focused on using different training algorithms by exploiting the structure of a single neural network model. Therefore, in this work, novel sequential training algorithms are proposed for the hybrid GRAB-NN models, which facilitates the independent estimation of RBF (i.e., c_{RBF} and σ_{RBF}) and ANN $\left(W_{hid_{ANN}}\right)$ parameters in the hidden layer, as well as the output layer weights $\left(W_{out_{GRABNN}}\right)$ by solving different optimization algorithms, while solving an outer layer optimization to ensure overall convergence.

The proposed sequential training algorithm considers the decomposition of the training objective function given by Eq. (4.47a) into three subproblems, represented as:

$$\min_{\boldsymbol{c_{RBF}},\boldsymbol{\sigma_{RBF}}} \boldsymbol{J_1} = \left(\boldsymbol{h_{RBF,tar}}(t) - \boldsymbol{h_{RBF}}(t)\right)^T R_1^{-1}\left(\boldsymbol{h_{RBF,tar}}(t) - \boldsymbol{h_{RBF}}(t)\right) \quad (4.48a)$$

$$\min_{\boldsymbol{W_{hid_{ANN}}}} \boldsymbol{J_2} = \left(\boldsymbol{h_{ANN,tar}}(t) - \boldsymbol{h_{ANN}}(t)\right)^T R_2^{-1}\left(\boldsymbol{h_{ANN,tar}}(t) - \boldsymbol{h_{ANN}}(t)\right) \quad (4.48b)$$

$$\min_{\boldsymbol{W_{out_{GRABNN}}}} \boldsymbol{J_3} = \left(\boldsymbol{y_{tar}}(t) - \boldsymbol{y_{GRABNN}}(t)\right)^T R_3^{-1}\left(\boldsymbol{y_{tar}}(t) - \boldsymbol{y_{GRABNN}}(t)\right) \quad (4.48c)$$

where, $\boldsymbol{h_{RBF,tar}}(t)$ and $\boldsymbol{h_{ANN,tar}}(t)$ refer to the target outputs from the hidden layer of the GRAB-NN model for the RBF and ANN nodes respectively.

The sequential parameter estimation approach proposed here also leverages the use of different optimization algorithms for solving each of the subproblems given by Eqs. (4.48a), (4.48a), and (4.48a).While the estimation of the RBF center (c_{RBF}) and width (σ_{RBF}) coordinates can still be accomplished through IPOPT / Adam SGD approaches discussed before based on whether the centers follow a deterministic or stochastic distribution respectively, the resulting optimization subproblem is considerably less complex, thus leading to much faster computation as compared to the simultaneous approach defined in Eq. (4.47a). In addition, the optimal hidden layer weights for the ANN nodes, i.e., $W_{hid_{ANN}}$ can readily be estimated by solving Eq. (4.48a) using various standard first-order backpropagation based algorithms such as gradient descent, scaled conjugate gradient, quasi-Newton method with line search, and Levenberg-Marquardt algorithms[226]. Furthermore, the linear representation of the output layer for the overall GRAB-NN model can also facilitate the implementation of OLS approach for estimating the corresponding optimal weight parameters $\left(W_{out_{GRABNN}}\right)$.

It is worth mentioning that the size of the hidden layer, i.e., number of neurons in the hidden layer (n_h) of the GRAB-NN model combining ANN and RBF nodes, can either be fixed at some pre-defined value or determined through an outer-layer optimization that optimizes the integer variable, i.e., n_h using any of the two approaches proposed in Section 4.4.3. For a given value of n_h, all possible combinations of ANN and RBF nodes have been compared to determine the optimal architecture of the hybrid GRAB-NN model for different nonlinear transient chemical process systems. The cubic spline interpolation algorithm[330] has been considered to initialize the parameters for each combination of ANN $\left(n_{h_{ANN}}\right)$ and RBF $\left(n_{h_{RBF}}\right)$ hidden nodes during training of GRAB-NN models. The detailed sequential training algorithm for hybrid GRAB-NN models for a given number of hidden layer nodes (n_h) is discussed under Algorithm 2. It is to be noted that the two extreme combinations possible for a given value of n_h leads to the standalone pure RBF $(n_{h_{RBF}} = n_h)$ and pure ANN $(n_{h_{ANN}} = n_h)$ models, respectively, with n_h neurons in the hidden layer. While the training of the standalone pure RBF network model thus obtained can be accomplished by implementing either of the two approaches proposed in Section 4.4.3, the standalone ANN model may be trained using various commercially available neural network / deep learning packages such as MATLAB Neural Network Toolbox, TensorFlow, PyTorch, CasADi[331], etc. While the formulation of the iterative algorithm (Algorithm 2) becomes straightforward due to the presence of two types of activation functions and one single layer of splitting, additional algorithms may be proposed if the hidden layer(s) consists of more than two types of activation functions and / or the overall network structure consists of more hidden layers.

Algorithm 2: Sequential Training Algorithm for Hybrid GRAB-NN Models for a given n_h

Initialize: $\boldsymbol{n_{h_{RBF}}} = \mathbf{1}$

Set: $\boldsymbol{\theta} \equiv \left\{\boldsymbol{c_{RBF}}, \boldsymbol{\sigma_{RBF}}, \boldsymbol{W_{hid_{ANN}}}, \boldsymbol{W_{out_{GRABNN}}}\right\}$

while $\boldsymbol{n_{h_{RBF}}} < \boldsymbol{n_h}$:

  Evaluate: $\boldsymbol{n_{h_{ANN}}} = \boldsymbol{n_h} - \boldsymbol{n_{h_{RBF}}}$

  Initialize: $\boldsymbol{h_{RBF,tar}}(\boldsymbol{t}), \boldsymbol{h_{ANN,tar}}(\boldsymbol{t})$

  while $\boldsymbol{\theta}$ not converged **do**:

    Obtain: $\boldsymbol{h}(t) = \left[\boldsymbol{h_{RBF,tar}}(t) \quad \boldsymbol{h_{ANN,tar}}(t)\right]^T$ (Eq. (4.42))

    Estimate: $\boldsymbol{W_{out_{GRABNN}}}$ by solving Eq. (4.48a) using $\boldsymbol{h}(\boldsymbol{t})$ as inputs

Estimate: $\boldsymbol{c_{RBF}, \sigma_{RBF}}$ by solving Eq. (4.48a)
Estimate: $\boldsymbol{W_{hid_{ANN}}}$ by solving Eq. (4.48a)
Evaluate: $\boldsymbol{h_{RBF}(t)},\ \boldsymbol{h_{ANN}(t)}$
Set: $\boldsymbol{h_{RBF,tar}(t) = h_{RBF}(t)};\ \boldsymbol{h_{ANN,tar}(t) = h_{ANN}(t)}$
end while
$n_{h_{RBF}} = n_{h_{RBF}} + 1$
Calculate the value of overall objective function $(\boldsymbol{J})$ as in Eq. (4.47a)
Return $\boldsymbol{\theta}$ (optimal parameters)
end while
Return $\boldsymbol{\theta}$ (optimal parameters) and $n_{h_{RBF}}$ corresponding to minimum value of $\boldsymbol{J}$

Training of Constrained RBF / GRAB-NN Models

In addition to the development of optimal unconstrained RBF / GRAB-NN models by implementing the various approaches proposed above, training algorithms have also been developed in this work to impose linear constraints on these data-driven models during both inverse (training) and forward (validation) problems. Our previous work[168] discusses the development of physics-constrained ANN models that ensures the exact satisfaction of linear mass balance constraints for nonlinear steady-state and dynamic chemical process systems by solving an equality constrained nonlinear optimization problem. However, for both standalone RBF and hybrid GRAB-NN models proposed in this work, the specific network architectures can be exploited to impose linear mass (atom) balance constraints on the respective model outputs. While the rigorous approach developed for training unconstrained RBF models will still require the linear constraint equations to be augmented as additional constraints during the parameter estimation problem, both the approximate approach for standalone RBF networks as well as the sequential training algorithms for the hybrid GRAB-NN models can take advantage of the CLS approach to ensure exact satisfaction of the imposed constraints during parameter estimation. This section briefly discusses the approaches developed in this work for steady-state and dynamic modeling of nonlinear chemical process systems using constrained RBF / GRAB-NN models for a given number of hidden layer nodes (n_h).

- **Steady-State Modeling**

For simplicity, similar to the training algorithms proposed for the hybrid GRAB-NN models, both steady-state and dynamic modeling approaches developed in this work for constrained RBF / GRAB-NN models assume that the corresponding model structures are fixed, i.e., n_h is not optimized during parameter estimation. Obviously, the modeling approaches proposed in this section for constrained RBF / GRAB-NN models can readily be combined with either the BB&B based rigorous approach or the approximate approach to optimize the size of the hidden layer (n_h) in an outer layer, if necessary.

For most chemical processes, it is desired that certain mathematical relationships such as mass and energy balances are exactly conserved, and not only approximated[239]. For instance, the mass conservation laws can simply be expressed as linear elemental atom balance equations in terms of only the input and output boundary conditions[168]. Therefore, the connection weights in the linear output layers of both standalone RBF (i.e., $\boldsymbol{w}$) and hybrid GRAB-NN (i.e., $\boldsymbol{W_{out_{GRABNN}}}$) can be estimated by substituting the OLS approach by the CLS algorithm, thus leading to the developed of steady-state mass-constrained RBF (McRBF) and mass-constrained GRAB-NN (McGRAB-NN) models respectively. The atom balance constraints

proposed for the McRBF and McGRAB-NN models involving input and output variables denoted as $\boldsymbol{x}$ and $\boldsymbol{y_{McRBF}}$ / $\boldsymbol{y_{McGRABNN}}$ respectively for the k^{th} steady-state observation index is given by:

$$\sum(\gamma_j * \boldsymbol{x}(k)) = \sum(\gamma_j * \boldsymbol{y_{McRBF}}(k)) \quad (4.49a)$$

$$\sum(\gamma_j * \boldsymbol{x}(k)) = \sum(\gamma_j * \boldsymbol{y_{McGRABNN}}(k)) \quad (4.49b)$$

where, γ_j represents the number of atoms of element 'j' with respect to the model input and output variables. The generic formulation of the CLS algorithm has to be appropriately modified based on the specific formulation of the linear constraints imposed on the training problem, in combination with the model equations for the proposed RBF / GRAB-NN structures.

- **Dynamic Modeling**

Though conventional feedforward data-driven models such as ANN, RBF, etc. show satisfactory predictive capabilities while modeling various steady-state systems, such networks may lead to considerable difficulties to accurately represent complex nonlinear transient systems with time-lagged input variables[180]. For effectively modeling such systems, hybrid architectures combining nonlinear static and dynamic network models developed in one of our previous publications[166] can be leveraged. These models have been shown to typically outperform several state-of-the-art deep recurrent networks, especially for large-sized complex nonlinear temporal training data. More details on the hybrid all-nonlinear static-dynamic neural network models developed for data-driven modeling of nonlinear process systems can be found in numerous references[166,168]. The steady-state modeling implementation of constrained RBF / GRAB-NN models can readily be extended for dynamic time-series data while taking advantage of the hybrid network structures.

Furthermore, it is worth mentioning that all different model structures considered in this work are fully data-driven models that can be developed if the input / output boundary conditions (measurements) are known even in absence of a rigorous understanding of the underlying process. Therefore, the formulation of appropriate constraint equations will vary based on the availability of sufficient holdup information. For many systems, measurement of materials holdup, directly or indirectly, may or may not be available during the development of data-driven models. The availability of measurements quantifying the accumulation of mass within the system is necessary if the proposed approaches aim to exactly satisfy mass conservation even during transients (in addition to steady-states). However, the absence of such holdup measurements for any given chemical process does not necessarily imply that the proposed McRBF / McGRAB-NN models cannot be applied for modeling those systems. It simply means that the mass balance constraints can only be applied at steady state during solving the training and simulation problems.

In this work, it has been assumed that the overall dynamic (time-series) data can be distinctly partitioned into steady-state and transient zones [332–335]. Motivated by the hybrid parallel all-nonlinear static-dynamic neural network architecture developed in one of the previous publications[166] by the same authors and under the assumption of distinct partitioning of training data, the steady-state McRBF / McGRAB-NN models are arranged in parallel to a deterministic nonlinear autoregressive type of recurrent neural network with exogenous inputs (NARX-type RNN), as shown in the block-oriented network

structure in Fig. 4.20. Network configuration for dynamic McRBF / McGRAB-NN models. The NARX-type RNN in the parallel structure, as shown in Fig. 4.20, represents a residual model for predicting the deviations obtained with respect to the transient data from the corresponding steady-states. While $\boldsymbol{y}(t)$ denotes the overall outputs from the dynamic McRBF / McGRAB-NN structure, $\boldsymbol{y}_{NARX}(t)$ refers to the outputs from the deterministic NARX-type RNN model. The steady-state zones, in the proposed algorithm, are characterized as continuous time durations within the overall temporal training dataset where the maximum deviation from the mean value does not exceed $\pm$0.5%.

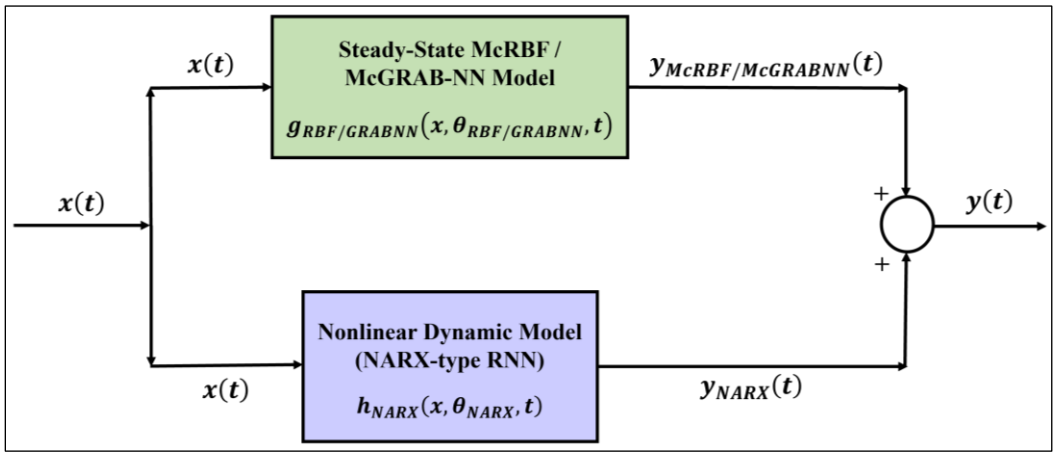


Fig. 4.20. Network configuration for dynamic McRBF / McGRAB-NN models

It is to be noted that the training approaches developed in this work can be implemented irrespective of the type of algorithm used for data partitioning. However, the partitioning of training data facilitates sequentially optimizing[166,168,170] the overall dynamic McRBF / McGRAB-NN models. In such an approach, the steady-state McRBF / McGRAB-NN models are first independently developed for exactly conserving the linear mass (atom) balance constraints at steady-state, followed by training of the nonlinear recurrent NARX-type deviation model by using the residuals obtained by calculating the difference between the training data and steady-state McRBF / McGRAB-NN model outputs. The deviation data thus generated serve as the target data for the NARX-type RNN model independently during the sequential approach while minimizing the typical squared error objective function as given by Eq.(4.50a). Similar algorithms for solving the various types of mass constrained neural networks during dynamic modeling can be found in a previous paper by the same authors[168].

$$\min_{\theta_{NARX}} \left(\boldsymbol{y_{tar}}(t) - \boldsymbol{y_{McRBF/McGRABNN}}(t) - \boldsymbol{y}_{NARX}(t)\right)^T R_4^{-1} \left(\boldsymbol{y_{tar}}(t) - \boldsymbol{y_{McRBF/McGRABNN}}(t) - \boldsymbol{y}_{NARX}(t)\right) \tag{4.50a}$$

$$s.t. \boldsymbol{y}_{NARX}(t) = \boldsymbol{h_{NARX}}(\boldsymbol{x}, \theta_{NARX}, t) \tag{4.50b}$$

4.4.4 Results and Discussions

Although the proposed standalone RBF and hybrid GRAB-NN models have been implemented on various nonlinear noisy transient chemical systems, including the industrial superheater system at the partner NGCC power plant, only the results obtained for an isothermal CSTR system have been discussed here for brevity. It has been specifically observed for the hybrid GRAB-NN models that the simultaneous parameter estimation and the proposed sequential approaches practically lead to the same optimal values of the model parameters corresponding to the same initial guesses, thus resulting in the same validation results for all case studies considered in this chapter. However, the sequential algorithms developed in this work result in significantly faster computation as compared to the simultaneous approach for each respective combination of ANN and RBF nodes in the hidden layer of hybrid GRAB-NN models. This, in turn, indicates that the proposed sequential training algorithms may be preferable in case the data-driven models need to be updated online. In addition, the rigorous approach for training standalone unconstrained RBF models consistently results in significantly lesser number of model parameters (stemming from lower values of n_h) compared to the approximate approach, as expected, due to simultaneous optimization of the coordinates of RBF centers / widths along with the connection weights in the output layer of the optimal networks.

For the isothermal CSTR system, the inlet volumetric flow rate (F), as well as the concentrations of all reaction species $\left(C_{A_f}, C_{B_f}, C_{C_f}, C_{D_f}\right)$ in the feed stream are considered as the model input variables for this case study, whereas, the outlet concentrations of all species (C_A, C_B, C_C, C_D) in the product stream obtained from the reactor represent the model output variables. In addition, the transient training data are generated by adding a non-zero time-invariant bias equal to approximately 5% of the mean of the data for each output variable to the simulated data, along with an additional Gaussian noise distribution $(\mu = 0, \sigma^2 = 0.25)$. It is to be noted that the principal motivation behind adding external bias / noise to the simulated data is to evaluate the performance of the proposed network structures and algorithms, especially those corresponding to the constrained RBF / GRAB-NN models, in presence of uncertainties in nonlinear time-varying training data. Furthermore, for all results presented for this case study example, only the validation plots showing the dynamic outlet concentration profiles of A (C_A) and B (C_B) are shown in this chapter. Other output variables show similar trends and therefore are omitted for brevity.

Results from Unconstrained RBF Models

The standalone unconstrained RBF models have been implemented to model the nonlinear dynamic Van de Vusse reactor using both the rigorous and approximate training approaches. It is observed that the optimal RBF models obtained by the rigorous approach consistently show improved model sparsity (i.e., significantly lesser number of model parameters) as compared to the approximate approach, since the former involves simultaneous estimation of the optimal coordinates of RBF centers and widths along with the connection weights in the output layer through an equality constrained optimization problem solved using Pyomo. However, it is to be noted that due to the associated MINLP (i.e., BB&B) approach, the rigorous training algorithm for especially large-sized unconstrained RBF models also leads to significantly higher computational expense than the approximate algorithm. Furthermore, for both types of training approaches, it has also been observed that the RBF model with probabilistic center distribution show relatively superior predictive capabilities as compared to that with deterministic centers for the same model size and structure. Fig. 4.21 shows the results obtained from the unconstrained RBF model with probabilistic centers using the rigorous training approach for validation data of the outlet concentrations

of A (C_A) and B (C_B) in the product stream. From Fig. 4.21, it can be observed that the RBF model under consideration accurately captures the overshoots and undershoots of the transient concentration profiles, leading to approximate validation RMSE and MAIE values of 1.5 $\frac{kmol}{m^3}$ and 4.3 $\frac{kmol}{m^3}$ for C_A, and 1.8 $\frac{kmol}{m^3}$ and 3.5 $\frac{kmol}{m^3}$ for C_B, respectively.

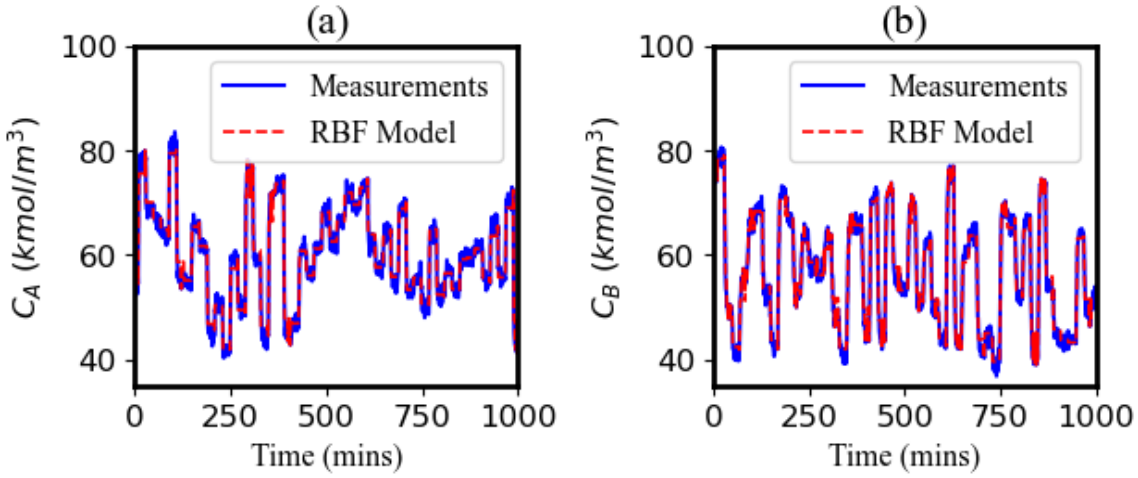


Fig. 4.21.: Results from the optimal unconstrained RBF model with probabilistic centers obtained through the rigorous training algorithm for validation data of (a) C_A and (b) C_B

Overall, considering all different output variables predicted by the optimal unconstrained RBF model with probabilistic centers, the RMSE and MAIE values consistently remain within 2% and 5%, respectively, for validation data of the Van de Vusse reactor system. Moreover, it is also worth mentioning that the BB&B approach significantly reduces the number of function evaluations while optimizing the network architecture without the necessity of evaluating all model parameters for each possible value of n_h (i.e., number of nodes in hidden layer), thus enhancing the computational efficiency of the rigorous approach. The comparison between the predictive performances of the optimal unconstrained RBF models with probabilistic and deterministic center distributions is provided in , along with the optimal model structure (i.e., number of hidden layer nodes as a measure of model sparsity) and the computational time during training of the corresponding models. It is seen that the training computational expense associated with the RBF model with deterministic centers for the rigorous approach is much higher compared to the RBF model with probabilistic distribution of centers. This difference in relative computational cost between the two types of network models is because the rigorous approach implemented on the unconstrained RBF model with deterministic centers involves optimizing the exact coordinates of all centers and width parameters under consideration through a second-order parameter estimation (i.e., IPOPT) algorithm. On the contrary, the rigorous approach applied for training the RBF models with stochastic centers simply updates the means and variances of the Gaussian distributions using a first-order approach (i.e., Adam SGD with mini-batch), from which the RBF centers (and widths) are sampled.

Table 4.7. Performance comparison between optimal unconstrained RBF models with deterministic / probabilistic centers using rigorous and approximate approaches for validation data of C_A and C_B

Types of Models	Training Algorithm	Validation RMSE $\left(\frac{kmol}{m^3}\right)$		Validation MAIE $\left(\frac{kmol}{m^3}\right)$		Number of RBF nodes in Hidden Layer (n_h)	Training Computational Time
		C_A	C_B	C_A	C_B		
Unconstrained RBF with Deterministic Centers	Rigorous	1.9	2.4	4.7	5.4	21	90 min
	Approximate	2.2	3.7	9.5	12.8	78	1 min
Unconstrained RBF with Probabilistic Centers	Rigorous	1.5	1.8	4.3	3.5	16	8 min
	Approximate	1.8	2.9	6.2	10.4	63	1 min

Table 4.7 also compares and contrasts the model validation results with respect to the rigorous and approximate training algorithms developed in this work while minimizing the objective function under consideration. While the optimal models obtained using the rigorous parameter estimation approach results in less than one-third the number of model parameters compared to the respective approximate approaches, the associated computational expense for the former is significantly higher than that of the latter. The validation plots showing the results obtained from the optimal unconstrained RBF models with deterministic and probabilistic center distributions using the approximate approach for parameter estimation for the dynamic validation data of C_A and C_B are included under **Fig. C. 8** and **Fig. C. 9** in **Appendix C**.

Results from Hybrid GRAB-NN Models

In addition to the unconstrained standalone RBF network models, the hybrid GRAB-NN structures have also been implemented to model the nonlinear transient outlet concentration profiles for the Van de Vusse reactor. However, in all results presented in this chapter for optimal hybrid GRAB-NN models, the overall model architecture (size) in terms of total number of hidden layer neurons, and hence the total number of model parameters, is kept constant to facilitate a fair comparison among different combinations of RBF and ANN hidden nodes with respect to the respective model predictive performances. It is to be noted that the proposed approaches can readily be extended to further optimize the number of ANN and/or RBF nodes in the hidden layer, and hence the overall architecture of the hybrid GRAB-NN models. The optimal value of $n_{h_{RBF}}$ may be estimated by an outer optimization loop incorporating the sequential BB&B approach proposed in this chapter for unconstrained RBF models. Similarly, the outer layer optimization to obtain the optimal number of ANN nodes in hidden layer (i.e., $n_{h_{ANN}}$) can be formulated using a sequential approach discussed in one of our previous publications[166]. The sequential algorithm for estimation of optimal number of hidden layer nodes in a typical three-layered ANN architecture[166] involves initialization of the network models with the minimum number of possible parameters based on the respective architecture, followed by sequentially increasing the number of model parameters after each iteration during training, thus leading to the generation of optimal parameters which minimize the specific objective function under consideration (such as in Eq. (4.44)). Moreover, in this work, since the overall architecture of the fully connected hybrid GRAB-NN model remains constant during estimation of the

optimal number of ANN and RBF nodes, the results presented in this section compare the relative values of mean squared error (MSE) calculated between the measurements and model predictions for each possible combination of network structure. Obviously, the variation in AICc follows the similar trend as compared to MSE because the last two terms in Eq. (4.44) remain the same for all combinations due to the exact same number of model parameters. Furthermore, the analysis is repeated thrice for each combination of network architectures (i.e., varying the number of ANN and RBF nodes in hidden layer keeping the total number of nodes constant) using different sets of initial guesses for the model parameters obtained by cubic spline interpolation. It has been observed that although the specific values of MSE obtained for each such analysis show similar magnitudes among different combinations, the trend in relative variation of MSE (and hence, AICc) remain exactly the same, thus leading to the same optimal model structure every time.

In this work, it has been assumed that the total number of nodes / neurons in the hidden layer of the hybrid GRAB-NN implemented for modeling the transient nonlinear isothermal Van de Vusse reactor system is given by:

$$n_h = n_{h_{ANN}} + n_{h_{RBF}} = 15 \tag{4.51}$$

As discussed before, the sequential and simultaneous parameter estimation algorithms developed in this work lead to the same optimal estimates of model parameters, however, the sequential approach is much more computationally efficient as compared to the simultaneous approach. For instance, in this case, the simultaneous approach converged in approximately 270 minutes (i.e., 4.5 hours) after evaluating all possible network structures and obtaining the optimal architecture, whereas, the proposed sequential algorithm recorded convergence in only around 10 minutes due to the flexibility of using different parameter estimation algorithms independently for various types of parameters (i.e., $\boldsymbol{c_{RBF}}, \boldsymbol{\sigma_{RBF}}, \boldsymbol{W_{hid_{ANN}}}, \boldsymbol{W_{out_{GRABNN}}}$) by exploiting the structure of the hybrid GRAB-NN model. It is to be noted that the two extreme combinations possible for $n_h = 15$ are the standalone three-layered ANN and RBF network models with 15 neurons in the hidden layer. It has been observed for both case studies considered in this work that all combinations of hybrid GRAB-NN model significantly outperform the standalone pure candidate models in terms of predictive capabilities for the same number of model parameters. Fig. 4.22 shows the variation in percentage MSE for all possible combinations of hybrid GRAB-NN models with $n_h = 15$ for validation data. It can be seen that the optimal architecture of the hybrid GRAB-NN model corresponds to the network structure with 10 RBF nodes and 5 ANN nodes in the hidden layer. It can also be seen from Fig. 4.22, that the standalone RBF model slightly outperforms the standalone ANN model in terms of validation MSE while modeling the nonlinear dynamic reactor system. The corresponding validation plots obtained from the optimal hybrid GRAB-NN structure while modeling the outlet concentration profiles of A (C_A) and B (C_B) are included under **Fig. C. 10** in **Appendix C**. It has been observed that the optimal GRAB-NN records validation RMSE values of around $1.2\,\frac{kmol}{m^3}$ and $1.5\,\frac{kmol}{m^3}$ respectively for C_A and C_B, while the corresponding MAIE values are $2.9\,\frac{kmol}{m^3}$ and $3.3\,\frac{kmol}{m^3}$ for the respective validation data.

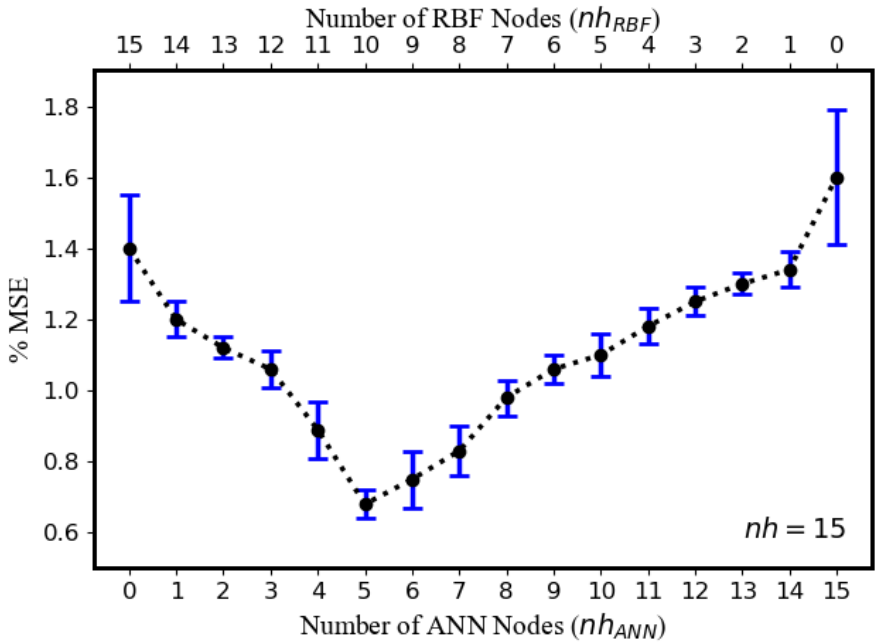


Fig. 4.22. Variation in average % MSE for all possible combinations of ANN and RBF hidden nodes in the hybrid GRAB-NN model for validation data ($n_h = 15$)

Results from McRBF / McGRAB-NN Models

In this section, the predictive performances of the optimal unconstrained standalone RBF and hybrid GRAB-NN models have been evaluated in presence of linear mass (atom) balance constraints for both steady-state and dynamic measurement data. Although the proposed approaches for exactly conserving overall mass of the system can be implemented for both RBF and GRAB-NN models (thus leading to McRBF / McGRAB-NN structures, respectively), the results presented in this chapter demonstrate the predictive capabilities of only the McRBF models for validation data of the nonlinear dynamic reactor system. The corresponding results obtained from the McGRAB-NN models also show similar variabilities with respect to the steady-state and dynamic data under consideration and hence, not included for brevity. Similar to the results discussed before, the optimization of the network structure (i.e., number of hidden layer nodes (n_h)) has not been considered while discussing the results obtained from the optimal McRBF models. Since the unconstrained RBF models with probabilistic center distribution have shown significantly superior predictive performance as compared to the RBF model with deterministic centers, the former type of RBF network architecture is considered for development of McRBF models. Furthermore, the approximate approach has been applied for estimating the coordinates of RBF centers and widths in the proposed McRBF architecture, for simplicity. However, the proposed training approaches for constrained RBF / GRAB-NN models can also be applied for RBF networks while optimizing the respective model parameters by an appropriate algorithm (IPOPT, Adam SGD, etc.) and / or considering n_h as an additional (integer) parameter to be estimated (i.e., the rigorous approach), as long as the external constraints imposed on the overall model are linear in model parameters.

For the CSTR system, it is assumed that the inlet and outlet volumetric flow rates are equal (i.e., $F_{in} = F_{out} = F$) across the reactor system. Moreover, since the reaction takes place in an aqueous solution, water (H_2O) is also considered as an additional species while formulating the mass (atom) balance constraints. Therefore, the linear mass conservation constraints for C, H, and O atoms, can be expressed

for this system by modifying Eq. (4.49a) as:

$$\sum_{\substack{i=A,B,C,D,H_2O \\ j=C\ (carbon),H,O}} \gamma_{j_i} * C_{i_f}(k) = \sum_{\substack{i=A,B,C,D,H_2O \\ j=C\ (carbon),H,O}} \gamma_{j_i} * C_{i_{McRBF}}(k) \quad \forall k = 1,2,\dots,k_n \tag{4.52}$$

where, k_n denotes the total number of steady-state observations available as training data. It is to be noted that for all results presented in this section showing error in mass conservation, an additional performance metric is considered, namely the absolute percentage error (APE) in mass (atom) balance. The APE for the i^{th} atom is defined as:

$$APE_{atom_i} = \left| \frac{\sum_{in} \dot{M}_{atom_i} - \sum_{out} \dot{M}_{atom_i}}{\sum_{in} \dot{M}_{atom_i}} \right| * 100\% \tag{4.53}$$

where, $\sum_{in} \dot{M}_{atom_i}$ and $\sum_{out} \dot{M}_{atom_i}$ represent the total mass flowrate of the i^{th} atom in all inlet and outlet streams respectively. For all results discussed in this chapter for this case study, plots are shown only for C_B. The other output variables show similar trends but are omitted for brevity. For the implementation of McRBF models on this case study, it is assumed that the total number of nodes / neurons in the hidden layer is given as $n_h = 25$.

Fig. 4.23 compares the model predictions, measurements, and truth, for the McRBF and standalone unconstrained RBF models for the validation data of C_B. The 'true' data is defined as the data generated by simulation of the process model for the isothermal Van de Vusse reactor system before the addition of external noise / bias to generate noisy measurements. It is to be noted that both the unconstrained RBF and McRBF models have been trained with respect to the noisy measurement (training) data. The primary objective of this study is to evaluate how the predictive performances of optimal McRBF models get influenced by the presence of noise in data in terms of exactly conserving the mass of the system. From Fig. 4.23 (a), it is evident that the predictions from the unconstrained RBF model show an excellent match with the noisy measurement data (represented by an RMSE value of around 0.8 $\frac{kmol}{m^3}$) while leading to larger mismatches with the system truth (as can be seen in Fig. 4.23 (c)), thus recording an approximate RMSE of 2.7 $\frac{kmol}{m^3}$. On the contrary, Fig. 4.23 (b) and Fig. 4.23 (d) show that the McRBF accurately captures the true data (recording an approximate RMSE of 0.7 $\frac{kmol}{m^3}$, while showing significantly larger errors (around 2.1 $\frac{kmol}{m^3}$) with the noisy training (measurement) data. In addition to the steady-state parity plots, Fig. 4.24 compares the McRBF and unconstrained RBF models in terms of exactly satisfying the C and H balance equations. As expected, the McRBF exactly conserves the linear atom balance constraints, while the corresponding values of APE are as high as 11% and 14% respectively for C and H atoms as obtained from the unconstrained RBF model. Similar plots for O balance as well as results obtained during model training have not been included in this chapter for brevity.

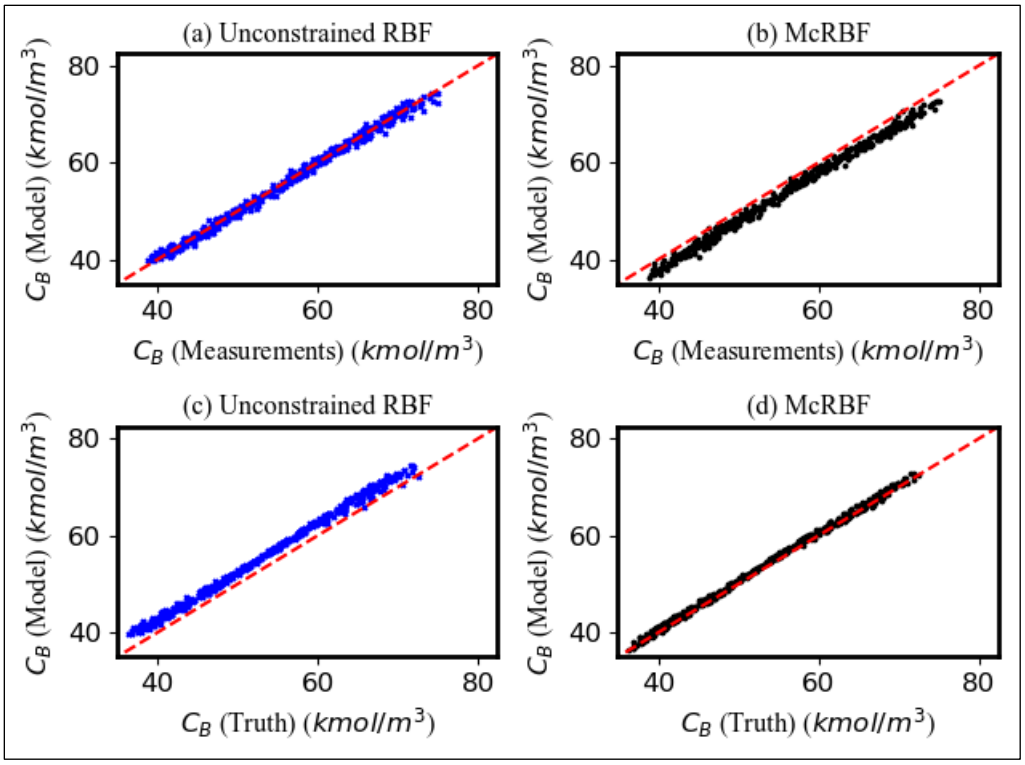


Fig. 4.23.: Comparison of results between McRBF and unconstrained RBF models with respect to true and measured validation data of C_B

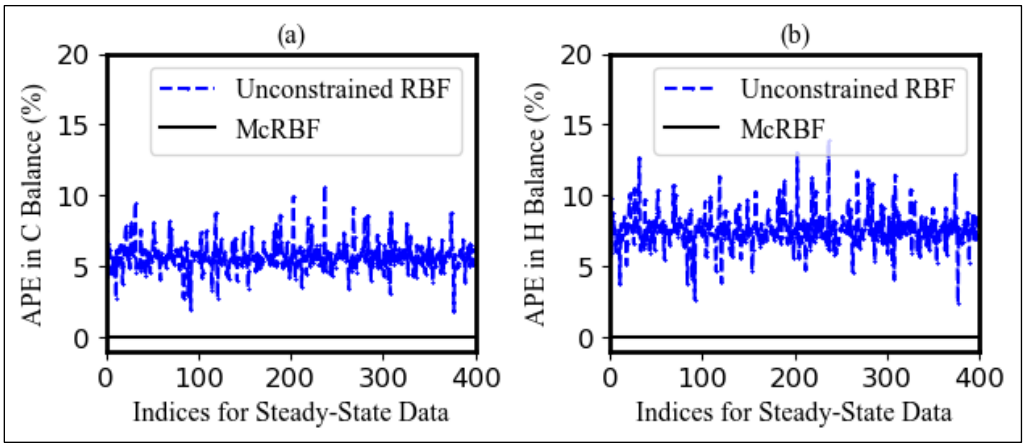


Fig. 4.24.: Comparison between McRBF and unconstrained RBF models in terms of violating (a) carbon (C) and (b) hydrogen (H) atom balance constraints during validation

For dynamic modeling implementation of the proposed McRBF models, the steady-state predictions obtained from the optimal McRBF model have been combined with a dynamic residual models aimed at estimating the deviations of the overall transient data with respect to the corresponding steady-states. The dynamic deviation neural network model represented by the NARX-type RNN has been trained using the widely used deep learning library, TensorFlow[336], in Python. In this case, the results of the dynamic McRBF (i.e., steady-state McRBF connected in parallel to the NARX-type RNN model) have been compared with the unconstrained network with a similar architecture, i.e., an unconstrained RBF model

connected in parallel to the NARX-type RNN. Fig. 4.25 shows the comparison between the model predictions versus truth as well as measurements for both constrained (i.e., McRBF) and unconstrained dynamic models for validation data of C_B.

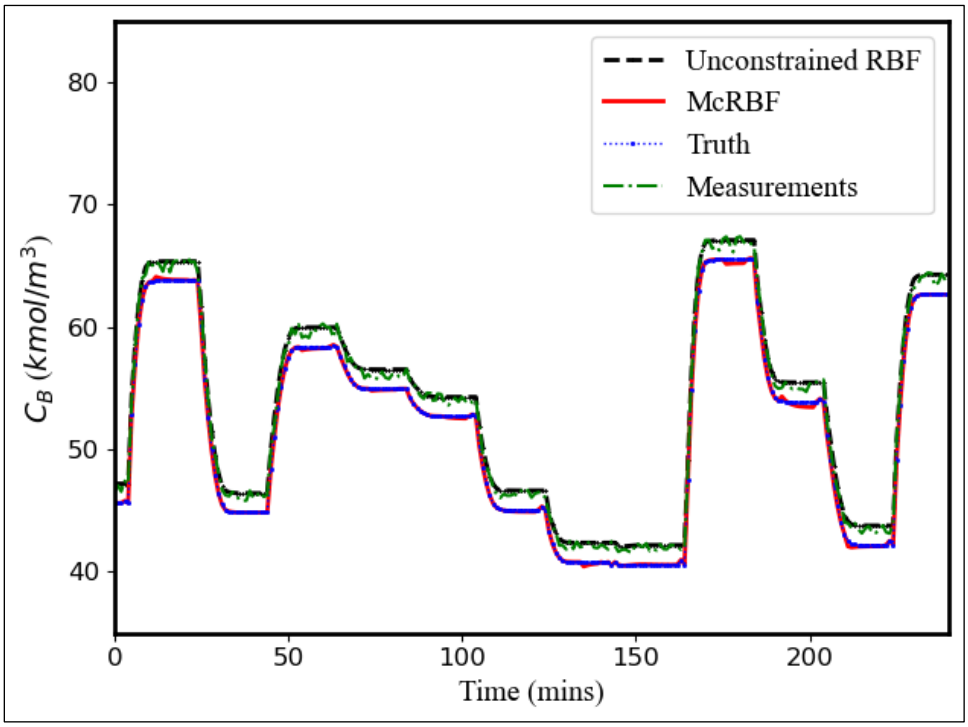


Fig. 4.25.: Comparison of results between dynamic McRBF and unconstrained RBF models with respect to true and measured validation data of C_B

It is to be noted that the proposed approaches for dynamic modeling can be applied irrespective of the specific type of dynamic / recurrent network architecture under consideration. As expected from Fig. 4.25, it is evident that the dynamic McRBF models yield results sufficiently closer to the system truth resulting in approximate RMSE value of 0.1 $\frac{kmol}{m^3}$ with respect to the true data, even when trained against biased / noisy training dynamic data. On the contrary, the unconstrained dynamic parallel structure combining the unconstrained RBF model with NARX-type RNN shows significantly larger RMSE (approximately around 1.6 $\frac{kmol}{m^3}$) with respect to the system truth, while recording excellent match with the measurements, thus leading to a corresponding RMSE value of 0.4 $\frac{kmol}{m^3}$. **Fig. C. 11** in **Appendix C** shows that the unconstrained model violates the mass balance constraints leading to a maximum of 23% and 31% APE in C and H balances respectively for validation data. In contrast, the dynamic McRBF models 'exactly' conserves system mass at steady state, while still showing some instantaneous errors during transients due to unavailability of system holdup information. It is worth mentioning here that appropriate formulation of linear (if possible) mass holdup constraints using only a subset of input / output boundary conditions may readily lead to incorporation of the same during model training in the CLS approach proposed in this work to ensure exact mass conservation at all time steps, instead of only the steady-state zones. Alternative parameter estimation algorithms for exact conservation of linear as well as nonlinear constraint equations in neural network structures during both training and forward problems have been extensively discussed in other works by the same authors[168].

4.4.5 Conclusions

The study focuses on the development of novel structures and training algorithms for standalone RBF as well as hybrid GRAB-NN models. In addition to the unconstrained model architecture, algorithmic capabilities have been developed to exactly conserve linear mass balance constraints during both training and forward problems, thus leading to the formulation of McRBF / McGRAB-NN structures for modeling nonlinear dynamic chemical process systems. It is observed that the rigorous approach for parameter estimation developed in this work using a BB&B approach results in enhanced model sparsity and prediction accuracy, but such training algorithms may incur excessive computational expenses. On the other hand, the approximate approach for training unconstrained RBF networks assigns the coordinates of the centers / widths a priori followed by estimation of output layer weights by the OLS / CLS approach based on the availability of appropriate linear set of constraint equations. For the case study, the optimal unconstrained RBF model with 16 hidden nodes consisting of a stochastic distribution of centers recorded validation RMSE values of 1.5 $kmol/m^3$ and 1.8 $kmol/m^3$ respectively for C_A and C_B using the rigorous training approach.

However, the optimal hybrid GRAB-NN model structure represented by a single hidden-layered network with 10 RBF and 5 ANN hidden nodes outperforms the optimal unconstrained RBF model, thus leading to around 1.2 $kmol/m^3$ and 1.5 $kmol/m^3$ RMSE for validation data of C_A and C_B, respectively. In addition, the inclusion of linear mass (atom) balance constraints on the model outputs for both steady-state and dynamic data ensures that the predictions obtained from the RBF / GRAB-NN models exactly conserve the mass of the system, even when trained against noisy measurement data that violate the same. It is observed that the McRBF models exactly satisfy the atom balance constraints at steady state while the unconstrained RBF models result in APE values in C and H balance as high as 11% and 14%, respectively, during model validation. In addition, the developed hybrid GRAB-NN models not only demonstrates the relative superiority of combining RBF and ANN hidden nodes on model predictive capabilities, but it also illustrates the importance of optimizing the structures of such networks (i.e., selection of optimal number of ANN and RBF nodes in this case) through the proposed MINLP algorithms. The authors also look forward to extending the proposed algorithms for incorporation of additional physics-based conservation constraints for chemical process systems.

4.4.6 Code Availability Statement

All data used in this section have been included in the form of figures and tables. The Python / Pyomo notebooks for development of different data-driven model structures and training algorithms proposed in this work can be accessed on GitHub via the following link: https://github.com/mukherjeeangan26/Hybrid_RBF_GRABNN

5 Development and validation of hybrid first principles-artificial intelligence models for boiler health monitoring

5.1 Introduction

Superheaters are critical components in power plants for producing the main steam at the desired temperature. Under load-following operation, these high temperature components can suffer from considerable damage that can eventually lead to failure and thus can compromise the reliability of the power plants. The work in this section investigates the use of hybrid models for spatio-temporal prediction of temperature in power plant superheaters.

Process systems can be modeled using first principles (FP) models based on conservation laws, material and thermodynamic properties, or using data-driven models based on measured operational data [337]. First-principles models can provide very good prediction even for cases where there are no or limited data availability in certain range of operating conditions or for cases where data acquisition is not feasible given the current state of measurement technology. However, constructing accurate FP physics-based mechanistic models for complex nonlinear noisy dynamic systems may lead to excessive computational expense as well as challenging for online adaptation. Furthermore, it can also be difficult, if not impossible, to develop efficient physics-based models for certain complex phenomena that are poorly understood. In contrast, artificial intelligence (AI)/machine learning (ML) models are typically much faster in computation and can be relatively easier to develop, simulate, and adapt online, even for complex and ill-defined dynamic systems[164]. One of the primary limitations of the conventional data-driven models stems from the lack of predictive or extrapolative capabilities especially when there is information gap in the data used for model development. For example, high-temperature systems such as steam superheater systems have complex dynamics with limited measurements for key state variables like tube metal wall and flue gas temperatures[338]. Modeling such systems using either a standalone physics-based or AI model can yield inaccurate results. Hybrid models, however, synergistically combine FP models with AI models and can be leveraged to yield a monitoring framework that is adaptive, real-time, and accurate[339]. This work develops several approaches where first-principles mechanistic models developed for high temperature power plant superheaters [147] have been synergistically coupled with two different ML models, i.e., artificial neural networks[166] (ANN) and Bayesian Machine Learning[234] (BML) based approaches, thus exploiting their key strengths[340].

Over the last few decades, hybrid FP+AI models have found numerous implementations[341,342]. One of the most common applications of hybrid models have been in the form of lumped parameter models where AI models account for certain static process parameters to be subsequently utilized in the FP model [343,344]. Other implementations of hybrid FP+AI models include modeling of the residual from a FP model by using data-driven models applied on chemical reactors[345], energy systems[346], and biological approaches[347], physics informed machine learning approaches aimed at conservation of system physics represented by mass[168,348], energy[349], and momentum balances[151], and digital twin development for coal fired power plants[350,351] and biomass energy systems[352]. These models can lead to improved interpretability and efficient extrapolation capabilities[353,354]. There are significant opportunities, however, for exploiting the synergistic hybridization of FP models with data-driven models. For such hybrid models to be successful and yield desired outcomes, various other aspects need to be considered other than coupling approaches. In particular, it is important to consider what information needs to be exchanged between FP models and data-driven models and at what interval, how

to select the specific data-driven model for the desired outcome, and when and how to adapt the hybrid model. Furthermore, if it is desired that both the FP and AI models represent nonlinear transients, then optimal synthesis of such hybrid models needs to consider complexity, computational expense, and accuracy.

For representing certain systems, more than one type of data-driven models may be simultaneously required in a hybrid framework[355,356]. The FP and data-driven models can be arranged in series or parallel or integrated or as combination of these configurations. If the FP model cannot adequately represent some phenomenon/phenomena of the process that is/are separable (i.e., neither the FP model impacts the results of the data-driven model, nor the data-driven model impacts the results of the FP model) and some or all of the inputs are used for both the FP and data-driven models, the parallel configuration is favored[357,358]. The deficiencies in the FP model may be directly expressed by a data-driven model as a function of model inputs[359]. For non-separable discrepancies where either the FP model outputs serve as additional inputs to the data-driven model(s) or the data-driven model outputs serve as the additional inputs to the FP model inputs, but not both, then a series configuration is preferred. Series (or serial) configurations of FP with data-driven models have shown improved extrapolation properties as compared to the parallel architecture[360]. For non-separable discrepancies, where the FP model outputs serve as the (additional) inputs to the data-driven model and vice versa, then an integrated configuration is preferred. In the integrated approach, the results from the FP model are affected due to expected improved estimates of parameters/states, etc. from the data-driven model. It should be noted that the results from the FP model are not affected in the parallel configuration or in series configuration if the FP model precedes the data-driven model. The integrated approach has been often used in the existing literature for estimating time-invariant parameters[343,344] by a data-driven model to be utilized in the FP model. If the parameters to be estimated by the data-driven models are time-varying, then the integrated approach can lead to challenging training and/or adaptation problems for dynamical systems. Obviously, various combinations of series, parallel, and integrated configurations may be desired depending on the discrepancies of the data-driven model and FP model and their parameters. Furthermore, in all such approaches discussed in the literature, where certain process parameters for the FP model are estimated by a data-driven model, the training data for the respective AI models are generated by a first-order error backpropagation-based approach. While typically such algorithms for regression / optimization are computationally faster, the resulting training dataset may lack sufficient richness due to local approximation and absence of accurate Hessian information[361]. Therefore, for online applications, the hybrid models thus developed may require frequent retraining to ensure satisfactory predictive capabilities for highly nonlinear transient process systems for prolonged continuous durations of operation.

The objective of the work in the section is to develop hybrid modeling approaches for predicting transient behavior of high temperature superheater components in power Plant A boilers. As the integration of renewables in the current energy grid is becoming prevalent worldwide to achieve low-carbon emissions, operation of existing fossil-fired power plants need to be flexible [362]. The current power generation fleet which is mainly designed for base load operations is facing several operational challenges due to these load-changing operations[363,364]. As a result, failure rate of the boilers especially that of the high temperature superheater components have gone up[69,70]. Advanced data-driven models utilizing extensive operational datasets have been well developed and utilized in industrial plants for rotary equipment such as gas and steam turbines especially for monitoring of those equipment items. For static equipment such as boilers, limited tools are available commercially. For developing a monitoring framework for these components, reasonably accurate estimates of temperature profile with sufficient spatial resolution are needed since the metal temperature affects creep and fatigue damages[365] in boiler

tubes and subsequent failure. However, it is challenging to develop accurate FP models for the superheaters due to multiple reasons. First of all, it is difficult to develop accurate models of the heat transfer characteristics in the superheater tubes due to build up of oxide scale inside the tubes [366]. The scale thickness keeps growing with time, but the scale can spall [367,368], which is highly stochastic. Furthermore, there is high uncertainty in the scale thermal characteristics [369,370]. In addition, in coal-fired power plants, there is deposit of coal fly ash on the outside of the tubes [58]. While frequent soot blowing is often used to clean those ash deposits, there may be fused ash that can be difficult to remove [371]. Thus, there is high uncertainty in the thickness and thermal characteristics of the ash deposit for coal-fired power plants. Second, modeling challenges for superheaters arise due to inhomogeneous spatial distribution of flue gas and steam flowrates, especially under load-following operation [372]. Due to the presence of thousands of tubes in this region and specific configuration of a given superheater, firing profile of the burners, etc., accurate dynamic modeling of flow inhomogeneity is challenging. Placing sensors in the superheater region is very challenging due to harsh operating conditions that not only make placement of sensors difficult [9,373], but life of the placed sensors are very limited especially in coal-fired boilers [374] where the superheater environment is highly erosive due to the fly ash-laden flue gas. Thus, developing hybrid models for the superheaters that can be eventually used for developing a monitoring framework is of high interest to utility operators.

Recently, a few hybrid modeling approaches have been implemented for process modeling, controls, and digital-twin development for thermal systems such as boiler and steam turbine for in-service power plants [350,375]. Other notable applications of hybrid models include tasks such as controlling fouling in biomass boilers [376] and performance enhancement of an ejector refrigeration system [377]. The hybrid models utilized in most of these works were a combination of lumped parameter physical mechanistic models and data-driven linear empirical models utilizing operational data. In particular, hybrid grey-box models have found implementations in industrial furnace systems [378], heat recovery steam generator (HRSG) [379] and various components of steam power plants (boiler, turbine, condenser, feed water system, etc.) [380,381]. The proposed models in these works generally used a combination of reduced order parametric models representing the physical system and static feedforward AI-based models for estimating unknown process parameters in thermal systems. Furthermore, grey-box modeling approaches for feedwater heater [382] and steam boiler system [383] have been developed by using detailed nonlinear FP differential algebraic equations (DAE) model based on mass and energy balances. The unknown system parameters were estimated for both the systems using nonlinear least-squares optimization utilizing operational measurement data. Similar approaches have been investigated for an electric hot water boiler system [384] and a superheater/reheater system in a combined cycle plant HRSG unit [260], using detailed mechanistic dynamic FP model based on differential mass and energy balance equations along with dynamic estimation of unknown heat transfer parameters. Dynamic hybrid models combining non-linear physics based models with gated recurrent unit (GRU) [346] or long-short term memory (LSTM) [385] type recurrent neural network (RNN) model have also been investigated for modeling heat absorption and fouling prediction in boiler systems. Recently, a hybrid modeling framework using dynamic models have been proposed for digital twin development in a coal-fired power plant [386].The proposed model structure consists of FP based discrete-time model for different sections of the plant such as pulverizers, furnace, drum etc. along with LSTM type of RNN models for estimation of model parameters. Although many of such hybrid structures utilizing dynamic AI models show satisfactory performance while representing critical process variables, the availability of limited transient operating data may lead to significant overfitting during model training, thus resulting in inferior predictive capabilities. Hybrid modeling techniques using lumped parameter physical models in addition to LSTM type RNNs [387,388] and fuzzy logic based AI approaches [25] have also been used for superheater steam

temperature control in industrial systems. More recently, a detailed dynamic distributed parameter model has been proposed for studying transient behavior of boiler systems under rapid load following operations [40,337]. The proposed models were validated using test data from an actual operating system showing its utility for simulating system behavior. Although most of the works referenced above have developed hybrid models of various components of thermal systems, most of these works have neither considered the spatial variability of process variables in the FP models nor the dynamics of time-varying parameters/variables that are represented by the AI models. Furthermore, a limited amount of operational dynamic data has been used for model validation in these works. An accurate prediction of the dynamic process variables and time-varying parameters for high temperature superheater systems is critical for developing a monitoring framework especially during load-following operations due to rapid changes in heat transfer performance and flow distribution during the transients.

The literature review on hybrid modeling of power plants identifies several gaps that we seek to fill in:

- Hybrid models of the superheater section of power plants using detailed distributed transient FP models along with dynamic data-driven models are currently not available.
- Modeling of time-varying parameters / variables by an AI model as part of the hybrid model for power plants has not currently been investigated. Moreover, the development of algorithms for generating training data during internal coupling between FP and AI models to address parametric uncertainties while ensuring sufficient data richness has not been explored.
- No hybrid model developed in existing literature provides estimates for the time-varying tube wall temperature profile in the superheater section of power plants.
- Main steam temperature is one of the most important variables for the power plant operation. A temperature that is higher than desired can cause significant damage to the steam turbine, while a temperature that is lower than desired can cause efficiency loss [5,260]. While the main steam temperature is a controlled variable and therefore measured, a predictive model can be helpful in developing a model-based efficient main steam temperature controller [5,260]. However, accurate prediction of the main steam temperature is difficult due to uncertain heat transfer characteristics, non-uniformity in flow direction, time-varying thermal holdup of the systems, etc. Hybrid models can be very useful for not only estimating the main steam temperature, but also other critical transient process variables such as metal wall temperatures of superheater tubes or flue gas temperatures, simultaneously. However, to the best of our knowledge, no work in existing literature discusses the development of a hybrid model for prediction of all these different types of transient temperature profiles (main steam, tube wall, flue gas) for pulverized coal (PC) or natural gas combined cycle plant (NGCC) power plants.
- Modeling spatio-temporal variability of temperature is critical for superheaters, where hybrid models can be particularly useful due to modeling and data challenges discussed before. We could not currently find any literature that has addressed this area.
- Model validation using experimental/industrial data has been generally done in the existing literature by using the steady-state data or dynamic data for a short operating range. It is desired that the models be validated in the expected range of power plant load and ramp rate as expected during the load-following operation.

In this section we present a systematic procedure for the selection of appropriate hybrid approaches for modeling of thermal systems in which dynamic industrial steam superheater system for a pulverized coal-fired (PC) boiler systems referred to as Plant A was considered as case study. In the present work, the FP models of both the industrial steam superheater systems which have been developed in our previous work [147] as two-dimensional (2-D) distributed DAE model considering differential mass and energy balance equations along with a detailed accounting of heat-transfer through the metal tube wall have been considered [389]. For the ML models to be used for online applications, it is important to take into consideration contrasting advantages, and disadvantages of various types of AI-based modeling techniques. As potential candidate data-driven models in the hybrid FP+AI framework discussed in existing literature, different variants of deep neural networks (DNNs) are typically considered, such as LSTM [387,390] or GRU [346] types of RNNs, as well as convolutional neural networks[391] (CNNs). However, such conventional DNN models comprise significantly large numbers of parameters, thus leading to potential overfitting during model training especially in absence of large datasets[191,192]. Alternative data-driven approaches commonly used in hybrid FP+AI models involve derivative-free algorithms such as fuzzy logic based decision making[25], etc. However, such approaches may suffer from slow convergence in absence of sufficient exploration and exploitation of the solution space due to absence of gradient information during parameter estimation. Both of these AI-based models / approaches prove to be considerably challenging especially for the power plant superheaters for which large sized dynamic input/output data may not be available for model development. Our previous works have shown that sparse deterministic and probabilistic data-driven models significantly outperform several state-of-the-art approaches while modeling complex nonlinear chemical systems from noisy dynamic data typical of real plants. Drawing from those studies, in this work, the deterministic data-driven models are represented by hybrid all-nonlinear series/parallel static-dynamic neural networks[166] (referred to as 'NN' for the rest of the report), while the probabilistic models are obtained from the BIDSAM (Bayesian Identification of Dynamic Sparse Algebraic Models) approach[234] (referred to as 'BML' for the rest of the report). These models/algorithms consider a significantly lesser number of parameters (for sparse system identification) as compared to typical DNNs and hence correspond to much faster computation if the hybrid model needs to be re-trained at frequent intervals during certain online applications. More details on the structures and training algorithms for these specific data-driven models can be found in our previous papers[166,234]. While the NN models typically ensure faster computation and accurate model fit for highly nonlinear transient data, the BML models offer capabilities for handling noisy data and uncertainty quantification along with estimation of interpretable model parameters.

5.2 Components of the Hybrid FP+AI Model

In this section, we provide brief discussions on the individual components of the hybrid models considered in this work. It may be noted that the various hybrid model components proposed in this work are generic and are flexible to be applied to model other complex nonlinear dynamic processes.

5.2.1 Dynamic Distributed DAE Model of the Superheater / Reheater (FP Model)

The dynamic first-principles superheater model used in the hybrid modeling framework is developed in our previous work [147]. The model used differential mass and energy balance equations for the flow streams along with consideration of metal tube wall temperature dynamics based on thick-walled assumptions. As illustrated in Fig. 5.1, the model consists of a control volume block approach. This provides a modular unit which can be combined in any form based on any industrial system geometry and configuration [392]. The approach can be easily extended to multiple superheaters/reheaters arranged in series /parallel.

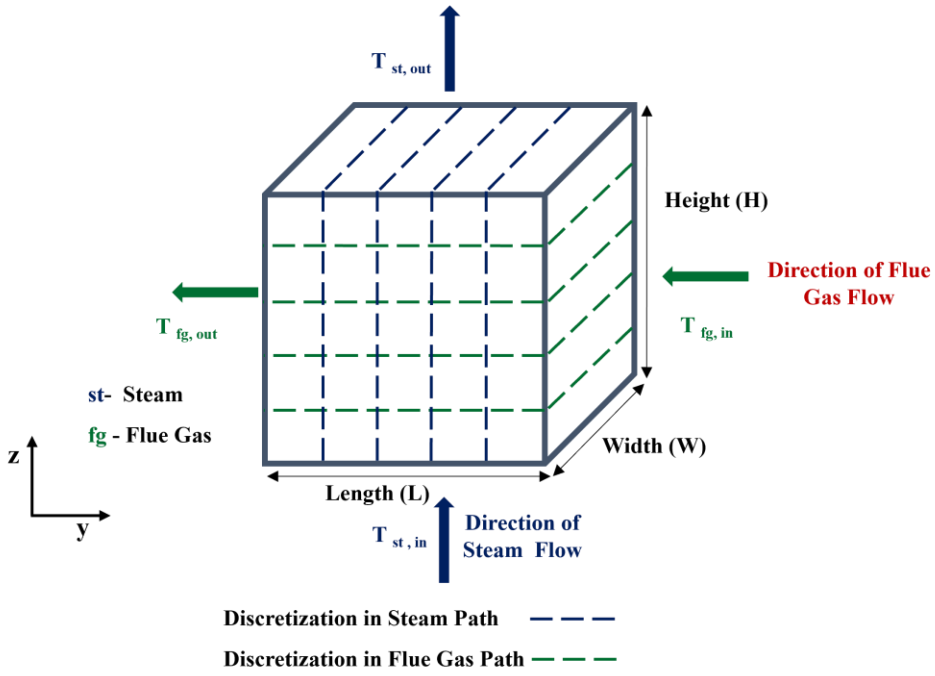


Fig. 5.1. Control volume block of the dynamic first principles superheater model.

In Fig. 5.1, a cross-flow arrangement is shown for the system with flue gas flow depicted by a green arrow flowing in the horizontal ‘y’ direction whereas steam flow is depicted by a blue arrow in the vertical ‘z’ direction. However, this considered flow pattern is generic and may be modified as per the system’s specifics. Also, the model is developed with the assumption that property changes for both the flow streams occur in their respective flow directions only. Hence, no change in the thermos-physical properties is considered in the width direction as shown in Fig. 5.1 for the two streams in the developed model. The differential equations are solved for each stream only in their respective flow direction neglecting the other directions. The dominant heat transfer mechanism considered for the model is convection only without any radiative effects.

The modeling equations for both the steam and flue gas are as follows:

Mass Balance Equations:

$$\frac{\partial \rho_{st}}{\partial t} + \frac{\partial (\rho_{st} v_{st})}{\partial z} = 0 \tag{5.1}$$

$$\frac{\partial \rho_{fg}}{\partial t} + \frac{\partial (\rho_{fg} v_{fg})}{\partial y} = 0 \tag{5.2}$$

Energy Balance Equations:

$$\frac{\partial T_{st}}{\partial t} + v_{st} \frac{\partial T_{st}}{\partial z} = + \frac{q_{wall,st}}{\rho_{st}\, c_{p,st}} \tag{5.3}$$

$$\frac{\partial T_{fg}}{\partial t} + v_{fg} \frac{\partial T_{fg}}{\partial y} = - \frac{q_{wall,fg}}{\rho_{fg}\, c_{p,fg}} \tag{5.4}$$

In addition to the mass and energy balance equations for steam and flue gas flow, the dynamic energy balance across the metal tube wall assuming thick-walled tube is also considered for the superheater system using the equation:

$$\rho_{tube} c_{p,tube} \frac{\partial T_{tube}}{\partial t} = k_{tube} \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T_{tube}}{\partial r} \right) \right] \tag{5.5}$$

where, ρ_{tube} and $c_{p,tube}$ are density and specific heat capacity of the metal tube wall, k_{tube} is the thermal conductivity of the tube metal wall and T_{tube} is the temperature of the metal wall.

The convective heat transfer on the steam side, considering the heat transfer from inside of the tube wall to the steam side, is:-

$$q_w = h_{conv,w} (T_{tube_{|r=r_{in}}} - T_w) \tag{5.6}$$

Similarly, the convective heat transfer on the flue gas side considering the heat transfer from the flue gas side to the outside of the tube wall is:-

$$q_{fg} = h_{conv,fg} \, (T_{fg} - T_{tube_{|r=r_{out}}}) \tag{5.7}$$

Here, q_w and q_{fg} are the heat flux, $h_{conv,w}$ and $h_{conv,fg}$ are the heat transfer coefficient and T_w and T_{fg} are the steam and flue gas temperatures for each control volume. $T_{tube_{|r=r_{in}}}$ is the inside tube temperature in contact with steam and $T_{tube_{|r=r_{out}}}$ is the outside tube temperature in contact with flue gas.

The boundary condition for the continuity of heat flux at the inside tube wall on the steam side is given by:-

$$q_w = (k_{tube} \frac{\partial T_{tube}}{\partial r})_{|r=r_{in}} \tag{5.8}$$

The boundary condition for the continuity of heat flux at the outside tube wall on the flue gas side is given by:-

$$q_{fg} = (k_{tube} \frac{\partial T_{tube}}{\partial r})_{|r=r_{out}} \tag{5.9}$$

Furthermore, the heat transfer coefficients and other associated parameters considered for the model are calculated using rigorous property correlations for both the steam and flue gas side as shown in Section 2.2 from Eq.(2.12) to Eq. (2.30). The steam side heat transfer coefficient is calculated using Gnielinski correlation for smooth tubes. The flue gas side heat transfer coefficient is calculated using Gnielinski correlation for flow across tube bundles. Non-uniformity at the inlet to the superheater on the flue gas side is taken into consideration by accounting for void fraction and tube arrangement factor based on inline and staggered tube arrangement.

5.2.2 All-Nonlinear Static-Dynamic Neural Networks (NN Model)

The all-nonlinear series and parallel static-dynamic NN models[166] considered in this work have been extensively discussed in Section 4.2, and hence not included here for brevity. Additionally, the codes for training and simulating the hybrid all-nonlinear static-dynamic network models are publicly available at: https://github.com/mukherjeeangan26/Hybrid_StaticDynamic_NN_Models.

5.2.3 Bayesian Identification of Dynamic Sparse Algebraic Models (BML Model)

Similarly, the BML algorithms used in this work have been thoroughly described in Chapter 3 and therefore, not included here for brevity. In addition, the codes for the BIDSAM algorithm for developing sparse models are also publicly available at: https://github.com/adeyesam/BIDSAM_Codes.

5.3 Configurations of Hybrid FP+AI Models

This section discusses each hybrid FP+AI model architecture proposed in this work and the motivation behind transfer of appropriate information to and from such FP and AI models while constructing accurate hybrid models for various dynamic process systems. It is to be noted that a particular hybrid model structure is neither specific nor exclusive to the corresponding implementation discussed in this report. Obviously, one or more structures proposed in this work can be applied to a particular objective subject to availability of necessary data. Although theoretically all structures are expected to be accurate for unanticipated operations, this work presents a systematic approach of selecting the optimal hybrid FP+AI model for representing key variables of interest.

5.3.1 Series Hybrid FP+AI Model

A possible configuration of the series hybrid FP+AI model is shown in Fig. 5.2 Obviously, the AI model can precede the FP model, or the FP model can be sandwiched between two AI models or other way round and there can be many such series of FP and AI models that can be considered. Additional synthetic inputs to the hybrid model may be the same $(\boldsymbol{u}_i(t))$ and/or different $(\boldsymbol{u}_{i,AI})$ as compared to the inputs of the FP model (refer Fig. 5.2). θ_{FP} and θ_{AI} refer to the model parameters for the FP and AI models, respectively, whereas $\boldsymbol{y}_{i,FPAI}(t)$ represent the final outputs from the overall hybrid model. It should be noted that for this specific structure of the hybrid model, the model structure, i.e., the arrangements and interactions between the FP and AI model, remain consistent during both training and validation (simulation) of the hybrid FP+AI models. This may not be the case for the other structures proposed in this work. The dynamic target outputs for the AI models are considered as the residuals $(\Delta_i(t))$ calculated between the measured values $\left(\boldsymbol{y}_{i,meas}(t)\right)$ and the FP model predicted values $\left(\boldsymbol{y}_{i,FP}(t)\right)$ of the i^{th} output variable, as defined by Eq. (6.7), for the final superheater systems under consideration.

$$\Delta_i(\mathrm{t}) = \boldsymbol{y}_{i,meas}(\mathrm{t}) - \boldsymbol{y}_{i,FP}(\mathrm{t}) \tag{5.10}$$

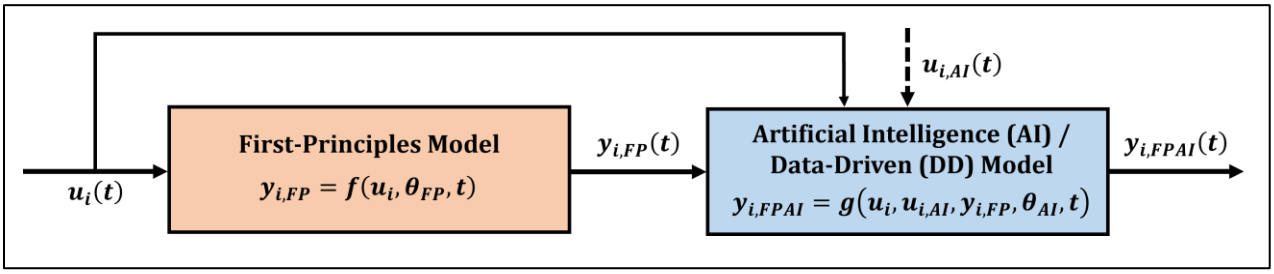


Fig. 5.2. Block-oriented configuration of series hybrid FP+AI model during training and simulation.

5.3.2 Integrated Hybrid FP+AI Model

The flow of information between the FP and AI models in the hybrid framework do not need to be one directional. For some applications, it may become necessary that information gets exchanged internally between the two models, as shown in Fig. 5.3. Such types of integrated cross-coupling configurations may be necessary to handle uncertainties in process parameters considered in the FP model. However, unlike the series hybrid FP+AI model discussed in Section 5.3.1, the integrated hybrid FP+AI model structure may slightly change during training versus simulation depending on what information gets exchanged from the FP to the AI models (represented by the dotted arrow in Fig. 5.3 (b). While a solid arrow in Fig. 5.3 (a) signifies that the transfer of information from the FP to AI models is mandatory during development (training) of optimal hybrid models, such transfer may not always be necessary during model simulation, as represented by the dotted arrow in Fig. 5.3 (b). For the case when the AI models utilize the instantaneous outputs of the standalone FP model as additional (synthetic) inputs during model training followed by subsequent transfer of information back to the FP model, the algorithm follows the same sequence even during simulation, i.e., when the optimally trained model is considered for future prediction subject to unknown inputs. In such examples, at every time step, the original FP model is solved first, followed by transferring partial or total information about the FP model outputs to the AI models. Thereafter, the data-

driven models utilize all information obtained from the FP model, along with additional inputs, if necessary, before sending back corrected estimates of certain parameters with which the FP model is simulated for a second time (within the same iteration/time index) leading to the final outputs $(\boldsymbol{y}_{i,FPAI})$ from the overall hybrid model.

The integrated arrangement of FP and AI models may also have certain specific applications, where the AI models do not require instantaneous estimates of any variable from the FP model as additional inputs during their training. In such examples, even though partial information from the FP model may get transferred to the AI models during training, such information is only utilized for data preparation for the AI models, i.e., not as additional inputs. Therefore, although the sequential integrated framework holds true during training/development of the hybrid model, the model architecture transforms to a series configuration during model simulation. At every time step during simulation of the optimal hybrid model, the AI models are first simulated followed by the FP model which can directly utilize the corrected parametric estimates transferred from the simulation of the optimal AI models. This second approach (i.e., considering $\boldsymbol{y}_{i,FP}$ only for data preparation for AI models instead of additional inputs) also reduces the number of function evaluations during forward simulation of the integrated hybrid FP+AI models when exposed to unknown inputs, since it involves the FP model to be solved only once, thus leading to relatively faster computation as compared to the first approach (i.e., considering $\boldsymbol{y}_{i,FP}$ as additional inputs for the AI models) where the FP model has to be solved twice during each iteration/time step.

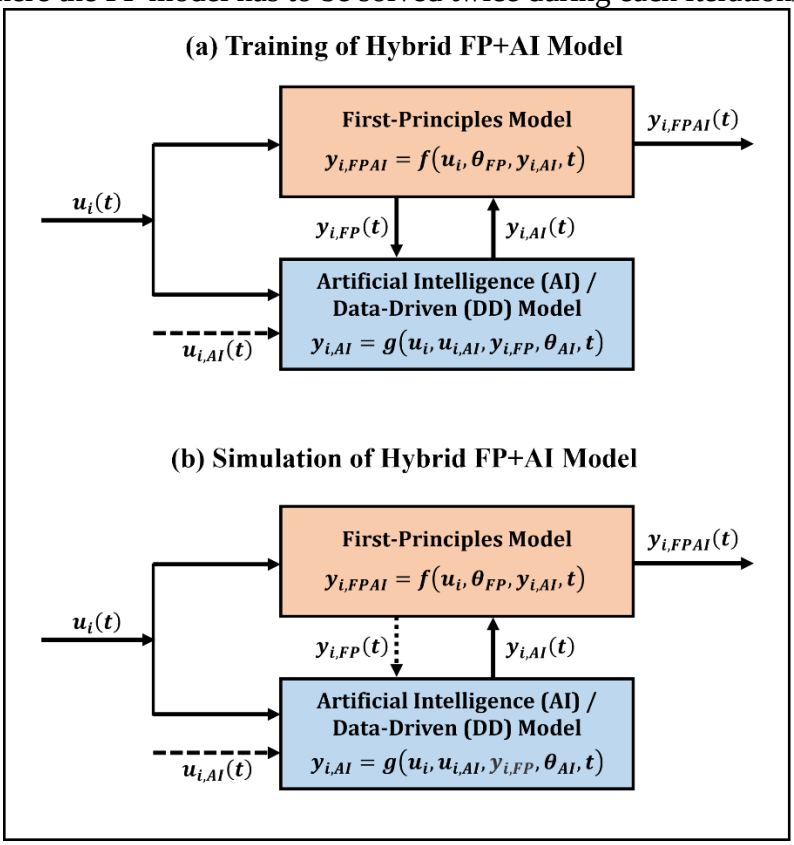


Fig. 5.3. Block-oriented configurations of integrated hybrid FP+AI model during (a) training and (b) simulation.

5.3.3 Parallel Hybrid FP+AI Model

The parallel arrangement of FP and AI models become necessary for various applications, where even though a mechanistic (FP) model is available, its predictive capabilities are limited due to inaccurate descriptions of certain effects, nonlinearities, or dynamic behavior. The fundamental difference of this type of configuration with the previous two structures described in Sections 5.3.1 and 5.3.2 is that the parallel hybrid FP+AI model provides the flexibility of optimal synthesis of both FP and AI models independently without requiring sequential transfer of any information from one to another or between each other, as shown in Fig. 5.4. However, partial or total information may be transferred from the FP to AI models during training the parallel hybrid FP+AI model (denoted by the dotted arrow in Fig. 5.4 (a), but such information will only be used in preparing input-output data for the data-driven models. The outputs of the data-driven models, for this configuration, act as additive corrections or augmentations of those obtained from the FP model, thus leading to the generation of the overall outputs from the parallel hybrid FP+AI model represented by $\boldsymbol{y}_{i,FPAI}(t)$.

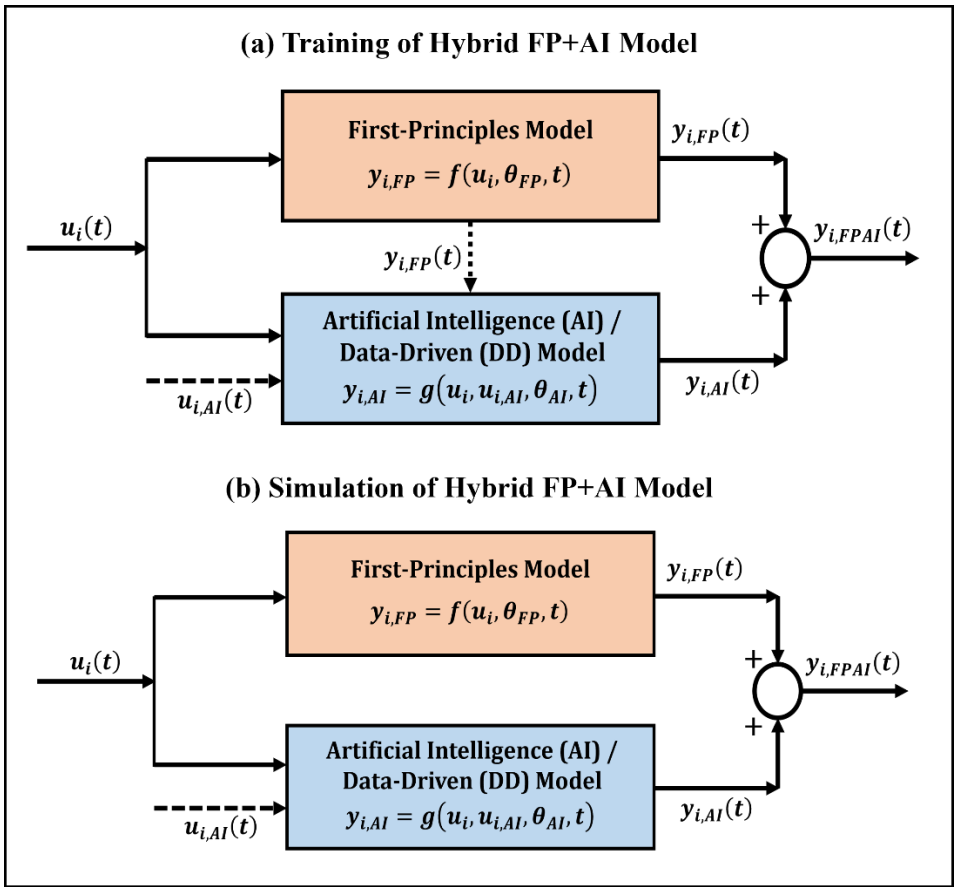


Fig. 5.4. Block-oriented configurations of parallel hybrid FP+AI model during (a) training and (b) simulation.

It is to be noted that depending on the specific application of the hybrid model, the parallel structure can accommodate various model forms substituting the FP and/or AI models. For example, the series hybrid FP+AI model (as described in Section 5.3.1) can readily replace the FP model in the parallel framework, thus leading to a 'series-parallel hybrid FP+AI model'. Similarly, if the FP block in the parallel structure gets substituted by the integrated hybrid FP+AI model (as discussed in Section 5.3.2), it may lead to the 'integrated-parallel hybrid FP+AI model' architecture. The corresponding architectures of series-parallel and integrated-parallel hybrid models during simulation have been shown in Fig. 5.5, where $\boldsymbol{y}_{i,FPAI_{ser}}$, $\boldsymbol{y}_{i,FPAI_{int}}$ and $\boldsymbol{y}_{i,FPAI_{par}}$ refer to the outputs from the series, integrated, and parallel hybrid FP+AI models, respectively. The training algorithms also follow similar structural representations.

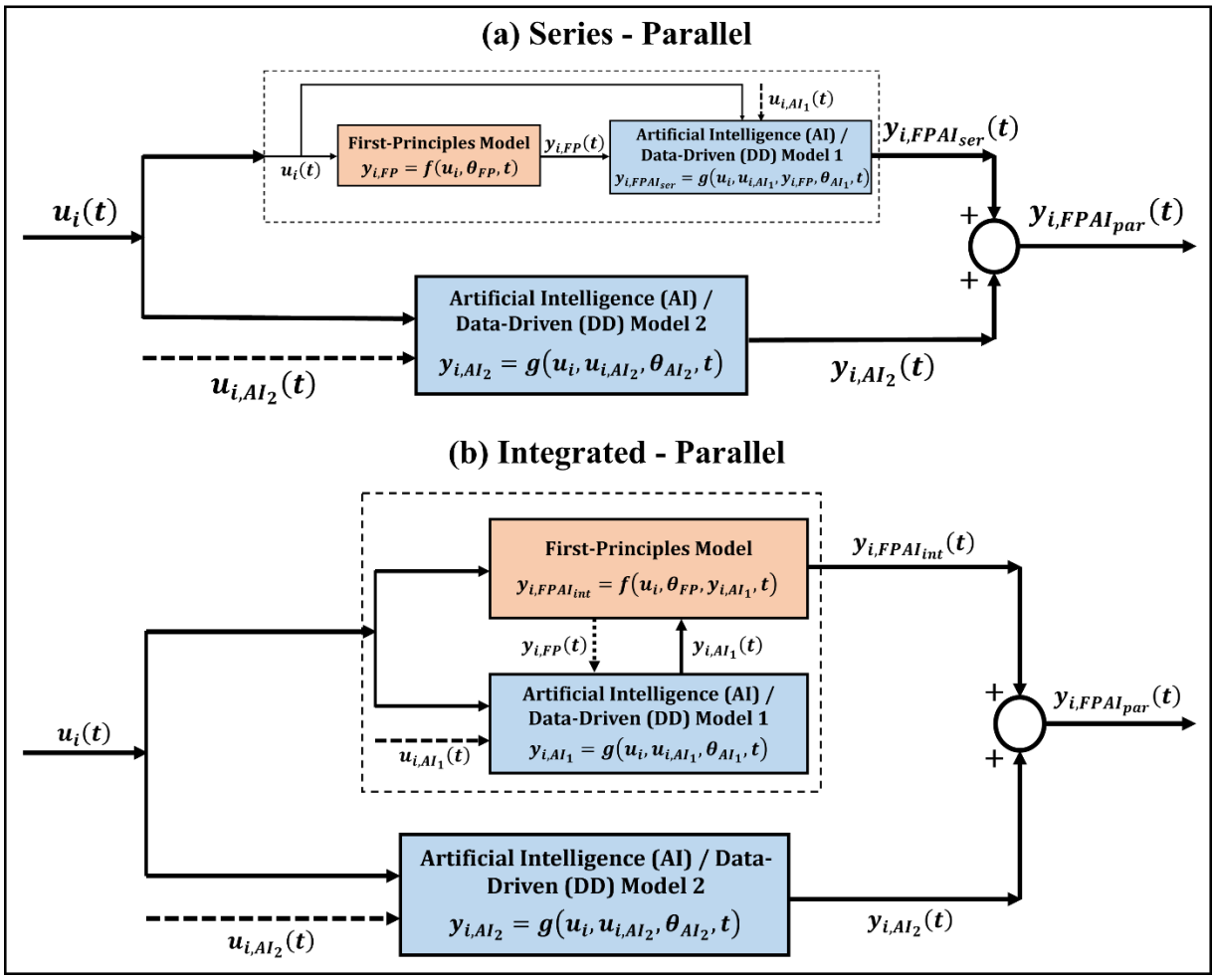


Fig. 5.5. Block-oriented configurations of (a) series-parallel and (b) integrated-parallel hybrid FP+AI models during simulation.

5.4 Description of Industrial Steam Superheater System (Plant A)

The final superheater system (Plant A) analyzed in this report is part of a PC-fired power Plant boiler with a generating capacity of 786 MW. The PC boiler fires bituminous coal via six pulverizers, producing main steam at nominal conditions of 537°C and 27 MPa. The configuration of the selected superheater system is shown in Fig. 5.6. The system consists of a two-pass tube assembly arrangement. Each pass of the given system consisted of six individual tubes, configured as a vertical outlet from the first header, with a 180 ^{o} smooth transition followed by a vertical section into the outlet header. It should be noted that the schematic diagram in Fig. 5.6 shows the transitions as angular rather than smooth connected to the steam inlet and outlet headers. The six individual tubes are numbered as T1 to T6 from right to left as shown in Fig. 5.6, with T1 being the longest and T6 the shortest in the flue gas path which flowed horizontally from right to left side as depicted in Fig. 5.6. Also, along the flue gas path, T1 is the first and last tube, while T6 tube is the innermost tube exposed to flue gas flow. The steam enters the system from the inlet header on the upper right-side section flowing down vertically with equal distribution into the six shaped tubes and exits the superheater system through the outlet header on the upper left-side section where all six tubes were welded. The outlet steam header connects further to the main steam line which connects to the HP steam turbine. The layout in Fig. 5.6 is the side view of the system. The entire final superheater system consisted of 197 identical flow assemblies, each with six U-tubes with a two-pass tube arrangement as shown in Fig. 5.6, stacked one behind another. This system was also uncharacteristically well instrumented, with the capability including 75 thermocouples in the penthouse attached to measure tubing across the width of the boiler aligned with selected assemblies' tube-to-header connections to the outlet header. However, the inlet temperature for the flue gas side (right side, as shown in Fig. 5.6) for this system is only available for a limited time duration from the plant historian and not for the entire duration which is needed for the model simulation. Hence, using the limited dataset, a quadratic correlation has been developed for representing the flue gas inlet temperature as a function of gross plant load (MW) and this correlation is then used for simulation of the entire dataset. It is observed that the data obtained for the superheater system from this PC-fired boiler system present more challenges as compared to that obtained from NGCC plants in terms of significantly higher systematic and random noise and considerable spatial distribution.

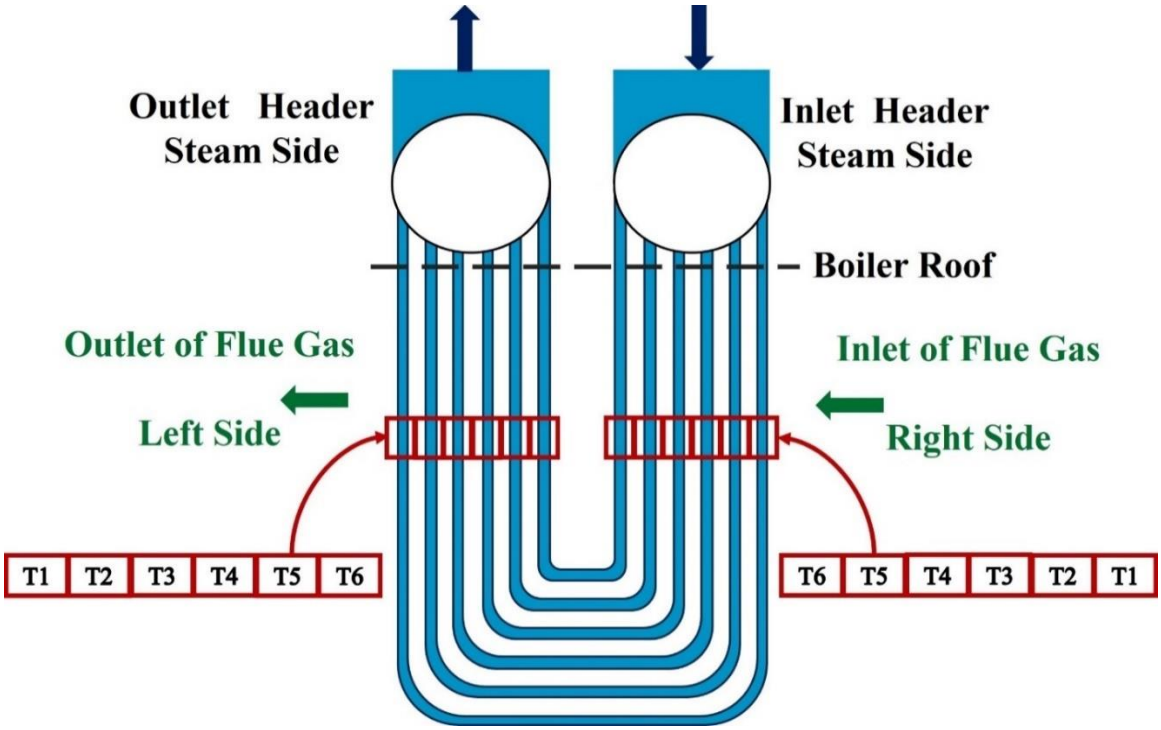


Fig. 5.6. Final superheater system (Plant A) layout. Note: the actual tube arrangement involves smooth transitions rather than the angular transitions shown in the schematic diagram.

The entire superheater systems for Plant A shown in Fig. 5.6 Is divided into individual control volumes as shown in Fig. 5.7 and the modeling equations from Eq. (5.1) to (5.9) are solved for each of these control volumes. Since the modeling equations were PDEs, the method of lines is used to convert these PDEs into ODEs. The reformulated ODEs are coupled with the algebraic equations making the entire system a DAE model. Fig. 5.7 shows the layout of both the superheater systems in control volumes and these entire systems have been solved in MATLAB using the DAE solver *'ode15s'*[393].

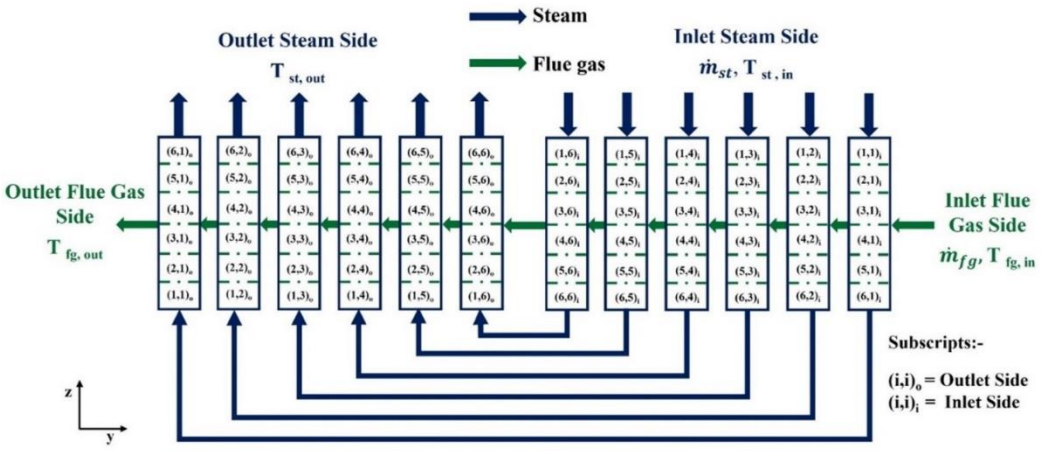


Fig. 5.7. Final superheater system layout using control volumes (second index of individual control volumes represents actual pass number of the superheater system) for Plant A.

5.5 Oxide scale growth study for Plant A

Model Development

The development of an oxide layer on the exposed tube surfaces inside the steam tubes is an inherent issue in boiler systems due to the deposition of iron or other oxides over time. The impact of these oxides' buildup is especially important in the superheater tubes with the highest average metal temperatures. Alloy steels having lower chromium content (e.g., T22 compared to T91) are expected to grow thick oxides over the course of their life. Depending on this steel composition, metal temperature becomes the primary accelerating factor for oxide growth. The buildup of these oxides components happens over an extended period, leading to decreased heat transfer and loss of efficiency. To study the effect of these phenomena, various growth laws can be used based on the temperature and exposure time for sections of the superheater system. In general, a parabolic growth scale is widely used to calculate the amount of oxide buildup over a period of time. The equation for the parabolic law considered for studying oxide growth in the final superheater system is as follow [394]:

$$d = [A * e^{\frac{-Q}{RT}} t]^{\frac{1}{2}} \tag{5.11}$$

where d is the oxide Thickness in micrometers (μm), T is the temperature in Kelvin, t is the time in hours, R is the gas constant, and A and Q are the oxidation parameters specific to the material of constriction(MOC) of tubes.

Furthermore, the development of the oxide scale on the inner tube surface over a period will decrease the heat transfer coefficient on the steam side, leading to an increase in the inner tube wall temperature. The effective heat transfer coefficient on the steam side is then calculated using the equation given below [395]

$$\frac{1}{h_{conveff,w}} = \frac{r_{in}}{k_o} ln \frac{r_{in}}{r_{in}-d} + \frac{r_{in}}{r_{in}-d} \frac{1}{h_{conv,w}} \tag{5.12}$$

where, $h_{conveff,w}$ is the equivalent Heat Transfer Coefficient (kW/m^{2}k) (Steam Side)

$h_{conv,w}$ is the initial heat Transfer Coefficient (kW/ m^{2}k) (Steam Side)

r_{in} *is* Tube's Inner Radius and k_o is the oxide Thermal Conductivity(kW/mk)

The final superheater system considered in the hybrid model validation work is utilized to estimate oxide growth buildup in its steam tubes. The configuration of the system is shown in Fig. 5.6. The tubes of this superheater system ate T22 alloy steel with the A and Q parameters for the parabolic growth law of Eq.(5.11) taken from literature [394].Since the growth of oxide is directly influenced by the increasing metal temperature. Hence, for the given study, we have considered predicting the oxide growth in the metal tubes near the steam outlet header for this system, as shown in Fig. 5.6 The reason for selecting the region near the steam outlet header is that it will operate very close to the main steam temperature. As a result, the metal tube surface near the steam outlet header will operate at higher temperatures than other superheater sections.

A comprehensive temperature dataset of about 2 years is obtained from the industrial data to study the effect of oxide buildup near the steam outlet header region using the installed Dutchmen at five tube locations. Hence a considerable long time of operating data has been considered. The material of construction (MOC) considered for the superheater tubes is T22 alloy steel with the oxidation parameter values for the parabolic growth law taken as-

$$\text{A} = 2.6\text{E}20\ \frac{\mu m^2}{h} \text{ and } \text{Q}=327\ \frac{KJ}{mol}$$

5.6 Results and Discussion

The performance of the different hybrid models formed by the synergistic integration of the standalone FP model with both types of AI models as discussed in Section 5.3 are evaluated by implementing them to improve model performance for critical variables such as main steam temperature for the industrial final steam superheater systems described in Section 5.4. For all case studies, the input and output variables of the data-driven models are normalized, and approximately 60% of the total dynamic data are used to train the corresponding AI (and hence, FP+AI) models. Cross validation is performed for both case studies using the k-fold method (with k = 5). For all structures and approaches the performances of different candidate models have been compared in terms of RMSE calculated between the measured output data $\left(\boldsymbol{y}_{i,meas}(t)\right)$ and model predictions $\left(\boldsymbol{y}_{i,mod}(t)\right)$, defined as:

$$MSE_i = \sqrt{\frac{1}{t_n}\left(\boldsymbol{y}_{i,meas}(t) - \boldsymbol{y}_{i,mod}(t)\right)^2} \tag{5.13}$$

where, t_n denotes the number of time steps under consideration. Furthermore, for developing accurate models of dynamic processes, especially energy systems, one of the key performance metrics is the instantaneous error[213] between actual measurements and model predictions. This measure does not get appropriately captured by RMSE or other similar measures. High instantaneous errors are unacceptable for a broad range of process applications, such as process monitoring, control, and dynamic optimization, that would utilize instantaneous predictions for decision-making. Therefore, significantly higher values of instantaneous error can eventually lead to various undesirable situations such as unsafe operation, equipment failure, poor product quality, and so on. For a measure of instantaneous error, in addition to RMSE, the MAIE is also calculated to compare prediction performances of the hybrid FP+AI models at any given instance. The MAIE for the i^{th} output variable is defined as:

$$MAIE_i = \max\left\{\left|\boldsymbol{y}_{i,meas}(t) - \boldsymbol{y}_{i,mod}(t)\right|\right\} \quad \forall\ t\ \in \{1,2,\dots,t_n\} \tag{5.14}$$

While results from the hybrid FP+AI models have been compared with the entire time-series temperature measurements that are available for the main steam for the superheater systems, only a subset of those results is included in this report for brevity. Furthermore, all comparisons regarding the relative computational expense discussed in this report correspond to that involved during training of the optimal data-driven models in the hybrid structures. During simulation, i.e., using the optimal hybrid FP+AI models for future time-series predictions, the AI models generate the respective output variables almost instantaneously. Thus, the simulation times associated with the hybrid FP+AI models depend mostly on the time the FP model takes to simulate through a single time step. In particular, the predictive capabilities of the proposed hybrid FP+AI models have been evaluated based on various performance metrics obtained while predicting the transient temperature profiles of the process variables of interest, i.e., the outlet steam, temperatures.

5.6.1 Results from Integrated Hybrid FP+AI Model

The final superheater system of Plant A (PC-fired boiler, refer Fig. 5.6) has been simulated and validated The control volume discretization scheme of Plant A is shown in Fig. 5.7 and inlet flowrates and temperatures of steam and flue gas obtained from plant operational data are considered as inputs to the standalone FP model. However, for this system, only the steam side input data, i.e., the steam inlet temperature $(T_{st,in})$ and steam mass flowrate ($\dot{m}_{st}$) are directly measured. The flue gas inlet temperature is calculated using a quadratic correlation for $T_{fg,in}$ relating flue gas inlet temperature with gross plant load (MW) for the entire dataset.
The quadratic correlation for relating inlet flue gas temperature with plant load is:

$$y = 0.0005 * x^2 + 0.2762 * x + 1543.5 \quad (5.15)$$

where, $x =$ Plant Load (MW), $y =$ Inlet Flue Gas Temperature (^{o}F) .

Similarly, the flue gas inlet mass flowrate ($\dot{m}_{fg}$) is also not available from plant data and has been calculated from combustion calculations using coal flowrates, air flowrates and coal ultimate analysis data obtained from the plant. Fig. 5.8 shows the profiles of all input variables used in the simulation of Plant A for a continuous period of 27 days. The selected dataset shows significant changes in unit operation with frequent load changes between minimum, intermediate, and maximum load with sufficient dynamic and steady zones captured in the dataset. The load profile of Plant A for the entire 27 days simulation period can be found in Fig. 5.9. Fig. 5.10shows the performance of the standalone FP model for predicting the main steam outlet temperature with respect to the measured data available from the plant historian for a continuous operation of 27 days. While the FP model generally captures the trend in the system dynamics, its predictions remain consistently biased to the measured plant data with an overall RMSE of 19.5 °C and a MAIE of around 35 °C. Both the RMSE and MAIE values obtained from the standalone FP model violate the desirable limits considered in this work. The overprediction by the FP model as reflected especially by the high MAIE value can be attributed to the model predicting higher heat transfer than observed in the actual system. The reason for this high heat transfer prediction by the model could possibly be due to certain modeling assumptions considered for the standalone FP model development impacting the accuracy of the FP model results such as considering convection as the dominant heat transfer mechanism ignoring the radiative heat transfer between flue gas and outer tube surface. Also, only the temperature dependence of specific heat capacity ($c_{p,st}$, refer Eq.(2.2)) for steam side is considered in the model development and not the pressure dependence along the steam flow path.

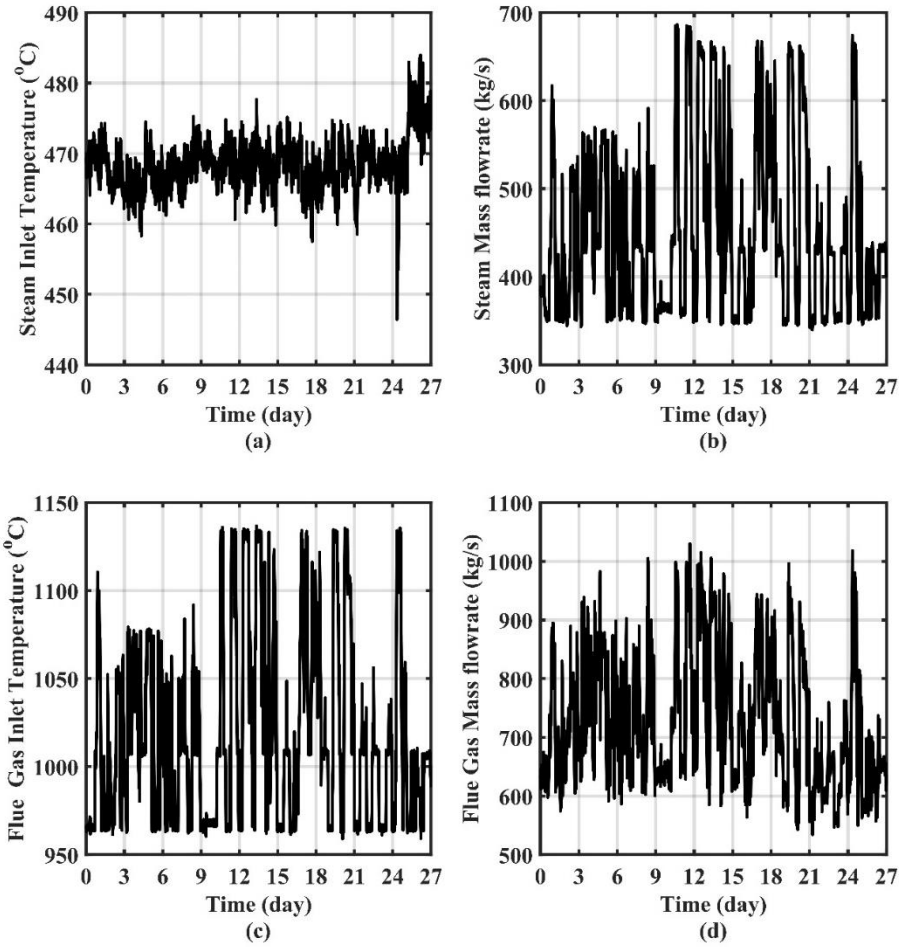


Fig. 5.8. Dynamic profiles of input variables for Plant A final superheater system considered for standalone FP model, i.e., (a) steam inlet temperature $(\boldsymbol{T_{st,in}})$, (b) steam mass flowrate ($\dot{\boldsymbol{m}}_{\boldsymbol{st}}$), (c) flue gas inlet temperature $(\boldsymbol{T_{fg,in}})$, and (d) flue gas mass flowrate ($\dot{\boldsymbol{m}}_{\boldsymbol{fg}}$) (see Fig. 5.7)

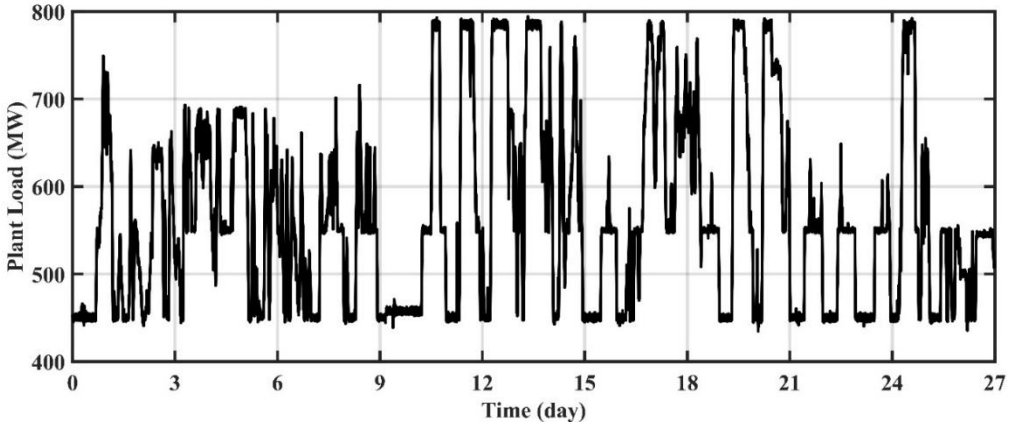


Fig. 5.9.. Temporal profiles of the plant load among low, medium, and high values for the superheater system at Plant A.

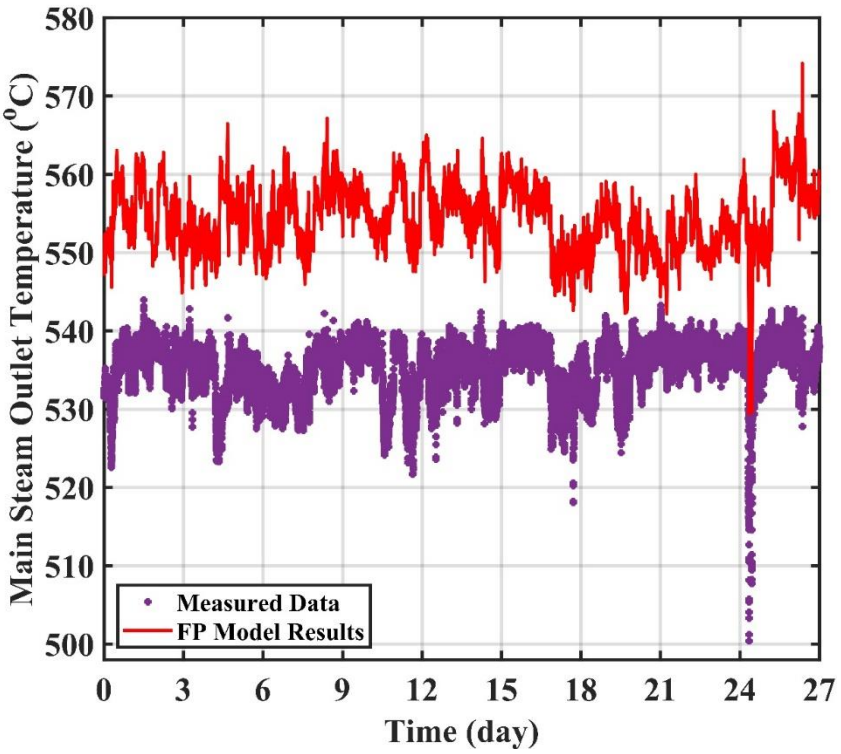


Fig. 5.10. Comparison of results from the standalone FP model with the measured data for main steam outlet temperature with respect to Plant A.

Modeling such energy systems can benefit from synergistic cross-coupling between the FP and AI models (i.e., integrated model as shown in Fig. 5.3). In the integrated hybrid model, the flue gas side heat flux (q_{fg}), given by Eq. (5.7) has been modified in the formulation of the FP model to account for the uncertainties in the heat transfer rate by incorporating an additive correction term for the heat transfer coefficient ($h_{corr,fg}$), as defined in Eq. (5.16). In Eq. (5.17), $h_{conv,fg}$ refers to the original heat transfer coefficient calculated from the heat transfer correlations involved in the FP model, and $T_{fg,in}$ and T_{tube} represent the inlet flue gas temperature and the outside tube wall temperature respectively.

$$q_{fg} = h_{conv,fg}\left(T_{fg,in} - T_{tube}\right) \tag{5.16}$$

$$q_{fg} = h_{conv,fg}\left(T_{fg,in} - T_{tube}\right) + h_{corr,fg}\left(T_{fg,in} - T_{tube}\right) \tag{5.17}$$

The corrective heat transfer coefficient $\left(h_{corr,fg}\right)$ values are represented by the AI models (refer Fig. 5.3 (a)). A continuous duration spanning around 27 days of plant operational data has been considered, out of which only 10 days (concatenating windows spanning for 24 h continuous time series data measured every minute, leading to approximately 33% of the total data) are chosen randomly which acts as a base model. For these data, a single parameter constrained optimization problem given by Eq. (5.16) is solved by using a second-order constrained optimization approach. The squared error between the results from the FP model, i.e., $T_{st,out}$ (refer Fig. 5.7), and the respective plant measurements is minimized by optimizing $h_{corr,fg}$.

$$\min_{h_{corr,fg}} \left(T_{st,out,meas}(t) - T_{st,out,FP}(t)\right)^2 \quad (5.18)$$
$$s.t.\ T_{st,out,FP} = f_{FP}\left(x, u, \theta_{FP}, h_{corr,fg}, t\right)$$

where, x denotes the state variables considered in the FP model and $f_{FP}(\cdot)$ denotes the system of FP model equations. The corrective term for the heat transfer coefficient ($h_{corr,fg}$) is expressed as a functional of the input boundary conditions (i.e., $T_{st,in}, \dot{m}_{st}, T_{fg,in}, \dot{m}_{fg}$) as model inputs. In addition to these four inputs, the attemperator spray flowrate $\left(\dot{m}_{spray}\right)$, as well as the valve openings on four inlet pipes connecting to the steam inlet header $\left(v_{p_A}, v_{p_B}, v_{p_C}, v_{p_D}\right)$, are considered as additional inputs for the AI models. The list of the input/output variables considered for the integrated hybrid FP+AI model can be found in Table 5.1.

Table 5.1. List of model input and output variables in the integrated hybrid FP+AI models for the superheater system of Plant A.

Variables used in Fig. 5.3	Actual input / output variables for the integrated hybrid FP+AI model in terms of process variables for Plant A
$\boldsymbol{u_i(t)}$	$T_{st,in}, \dot{m}_{st}, T_{fg,in}, \dot{m}_{fg}$
$\boldsymbol{y_{i,FP}(t)}$	$h_{corr_{opt},fg}$ (for data preparation during training) No $\boldsymbol{y}_{i,FP}(t)$ during simulation
$\boldsymbol{u_{i,AI}(t)}$	$\dot{m}_{spray}, v_{p_A}, v_{p_B}, v_{p_C}, v_{p_D}$
$\boldsymbol{y_{i,AI}(t)}$	$h_{corr,fg}$ (during simulation)
$\boldsymbol{y_{i,FPAI}(t)}$	$T_{st,out}$

It should be noted here that out of the five additional inputs considered for the AI models only four input which are the valve openings $\left(v_{p_A}, v_{p_B}, v_{p_C}, v_{p_D}\right)$ are directly measured in the plant using instruments. The attemperator spray flowrate $\left(\dot{m}_{spray}\right)$, although available from plant operating data is not a measured value and is calculated using a developed correlation by the Plant A based on valve openings and plant load. Fig. 5.11 shows the comparison between the integrated hybrid FP+NN and FP+BML models for the main steam outlet temperature in Plant A. It can be observed that the integrated hybrid FP+AI models accurately capture the overshoots and undershoots in the temporal profile of the main steam outlet temperature with desired accuracies.

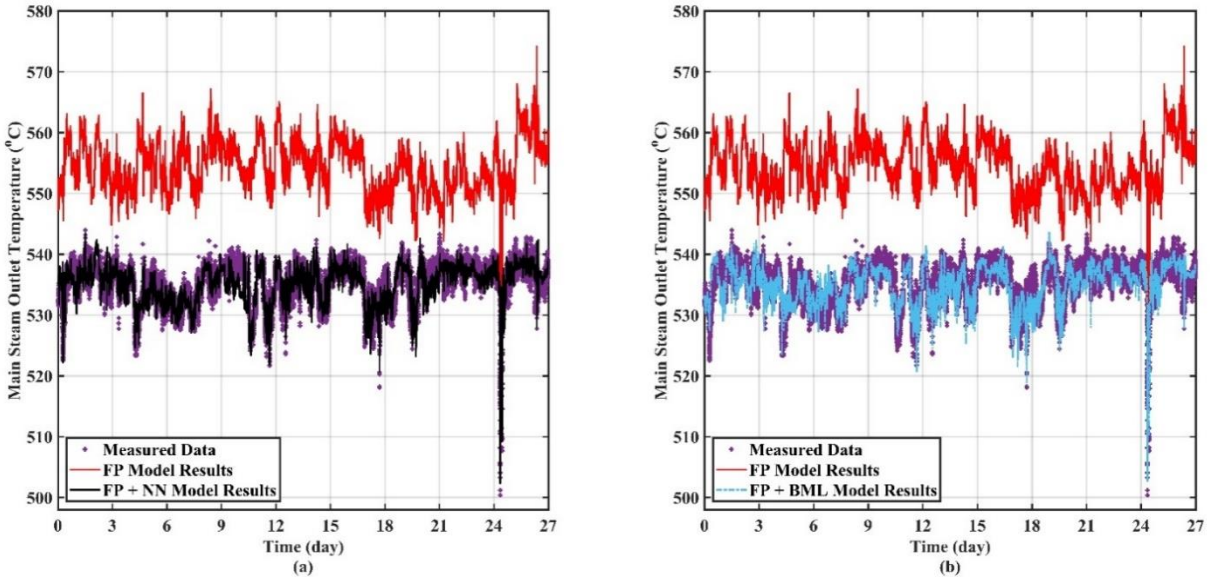


Fig. 5.11. Comparison of results between standalone FP vs hybrid (a) FP+NN and (b) FP+BML models for main steam outlet temperature with respect to the superheater system at Plant A.

The comparison among the different performance measures among the standalone FP, FP+NN, and FP+BML models has been provided in Table 5.2. Both types of hybrid FP+AI models significantly improve the prediction accuracy within the acceptable limits of RMSE and MAIE considered in this work, as can be seen in Table 5.2. It should be noted that the training computational times reported in Table 5.2 does not include the computational expense associated while generating the training data for the AI models by solving the constrained optimization problem.

Table 5.2. Comparison between hybrid integrated FP+NN and FP+BML models in terms of overall prediction RMSE, MAIE, number of parameters and training computational time for the main steam outlet temperature of Plant A

Types of Models	**RMSE (°C)**	**MAIE (°C)**	**Number of Model Parameters**	**Training Computational Time for the respective AI models**
FP + NN	1.9	6.5	100	1.7 min
FP + BML	2.0	8.5	27	22.9 min

From Table 5.2, it is evident that both hybrid integrated FP+AI models show similar predictive performance for the main steam outlet temperature. However, the BML model requires less than one-third of the number of parameters used in the NN model. It is to be mentioned here that for all case studies, consistently higher computational expense associated with the training of the BML model can be attributed to the rigorous search for the globally optimal subset of well-defined basis functions using the branch and bound approach in the BIDSAM algorithm [234]. In addition to this, the approach explicitly accounts for noise that may be present in the training data as it simultaneously carries out uncertainty quantification alongside model parameter estimation.

Now, using the base FP+AI (NN or BML) model developed for the 27 days period and whose validation results are shown in Fig. 5.11 , a period of 22 months is considered for main steam outlet temperature validation using the developed base FP + AI models. Fig. 5.12 compares the measured vs. hybrid model value for the steam outlet temperature for the given 2 -Pass Superheater system considering a large dataset of 22 months. In the given figure, the dashed black lines represent period of shutdown in the plant and the solid blue line is the base model data considered for the NN and BML training data. Although initially the model performed quite well in predicting the measurement data. However, after certain time especially after month 12 when no training data has been used for black box model training onwards a significant deterioration is observed between hybrid model value and actual data .Furthermore, in the last 4 months periods from month 18- 22 a large error is observed between model and measurement value indicating some operational changes in the plant which might have led to such scenario. In order to address this mismatch an adaptive training approach could be utilized further in the NN/BML model development.

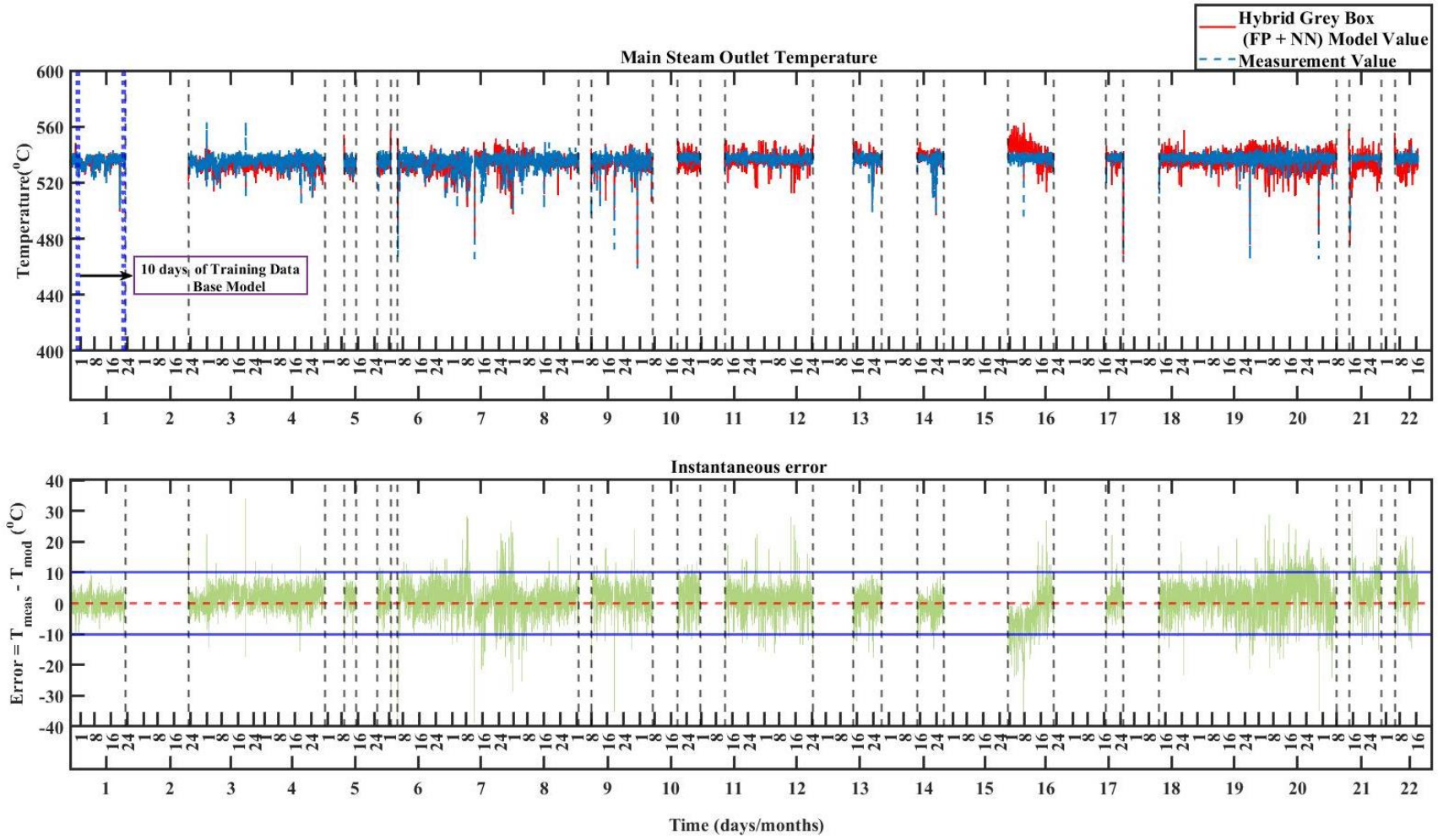


Fig. 5.12. Main steam outlet temperature validation using hybrid grey-box modeling (FP +NN) for 22 months using training data for 10 days

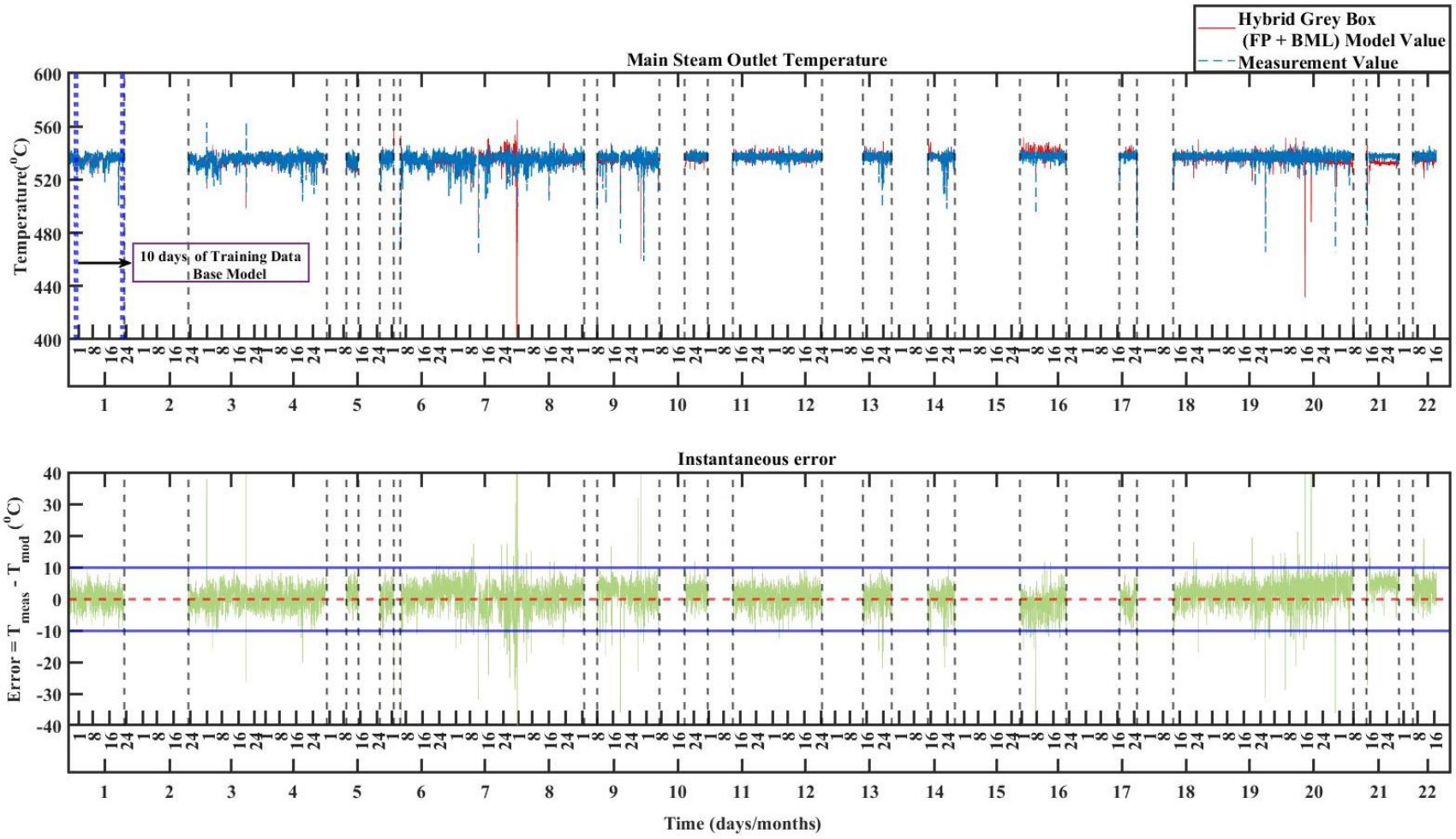


Fig. 5.13. Main steam outlet temperature validation using hybrid grey-box modeling (FP +BML) for 22 months using training data for 10 days

In Fig. 5.13, the similar hybrid model with BML model for the 22 months period has been shown. The results for the BML model also seem to correspond well with the ones obtained using NN model. However, for a certain time around month 7 the BML failed due to certain inputs going out of bounds while using the training data showing very large error. For the same time NN model was still working for that zone. However, the large error observed in month 18-22 between model and measured value for FP+ NN model seems to be less while using the BML model. This could be attributed to better model sparsity and interpretability offered by BML compared to NN. However, the large error seen at some places of the BML model can lead to significant mismatch and are not acceptable requiring an adaptive training approach for data generation.

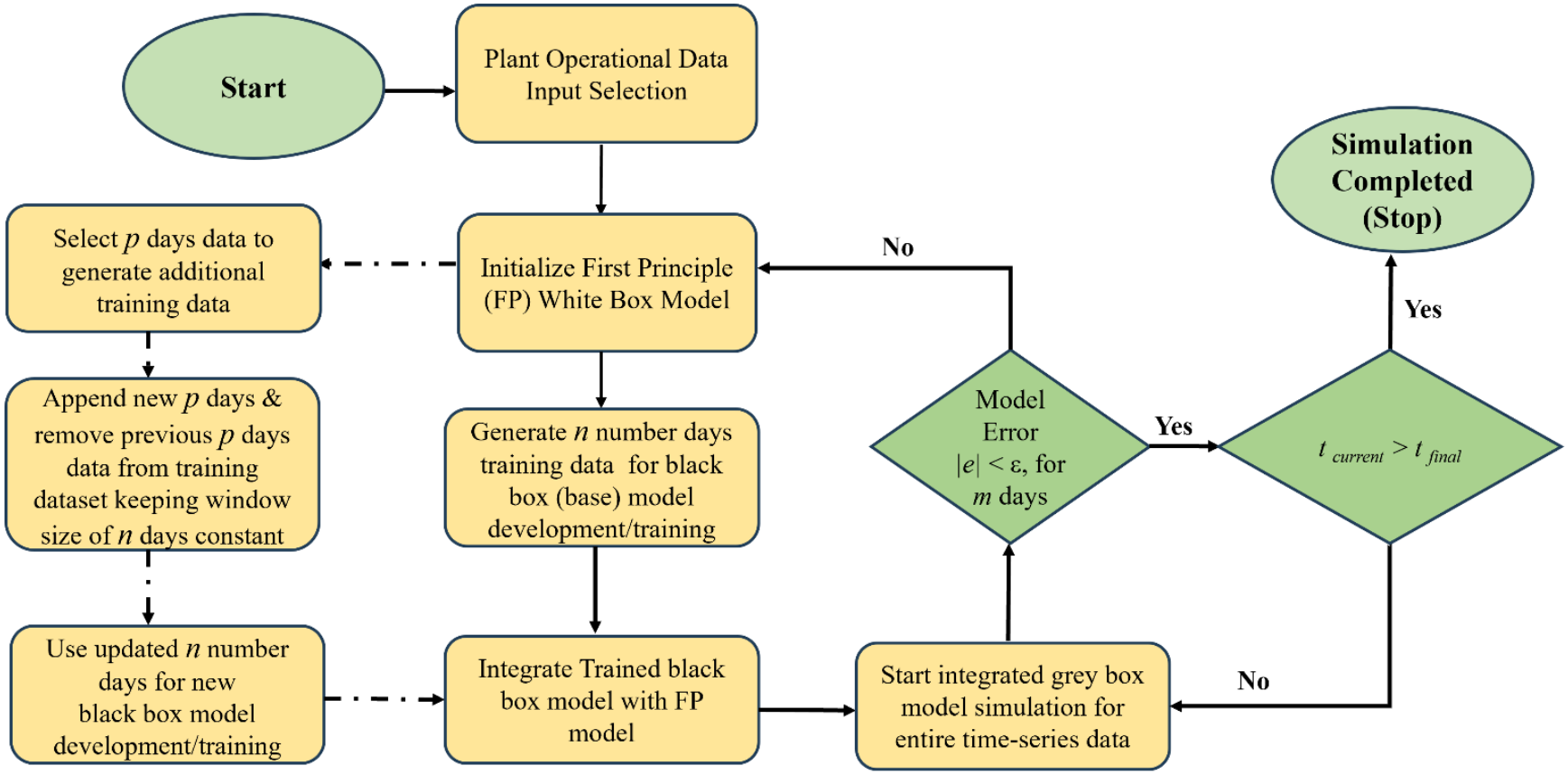


Fig. 5.14. Adaptive Hybrid Modeling algorithm

Fig. 5.14 shows the schematic flowchart for utilizing adaptive training for generating the dataset for the integrated FP+AI models. In the adaptive methodology, an initial base model is developed initially with some *n* days of data considered for generating training dataset. In our present case study, the base model is developed using 10 days of training dataset out of a period of 27 days as shown for the main steam outlet temperature validation in Fig. 5.11. The base model is then utilized for simulation of the remaining time series data for the hybrid models as shown in Fig. 5.12 and Fig. 5.13. In case a significant error is observed between the model and measurement data for a continuous period of *m* days then the simulation is stopped at that point and *p* days of data from the time period where the model started having significant deviation is taken and appended to base model training dataset by adding this new *p* days data and simultaneously removing old *p* days data from the base model dataset to keep the dataset window constant at *n* days. In this manner, a moving window approach has been adopted in which whenever model performance start deteriorating some additional new training dataset is appended to the training dataset window while removing a fixed amount of previous training dataset from the window keeping its size constant. The adaptive integrated hybrid FP+AI models for both FP+NN and FP + BML are simulated for the main steam outlet temperature for Plant A and the results are shown in Fig. 5.15 and Fig. 5.16 .

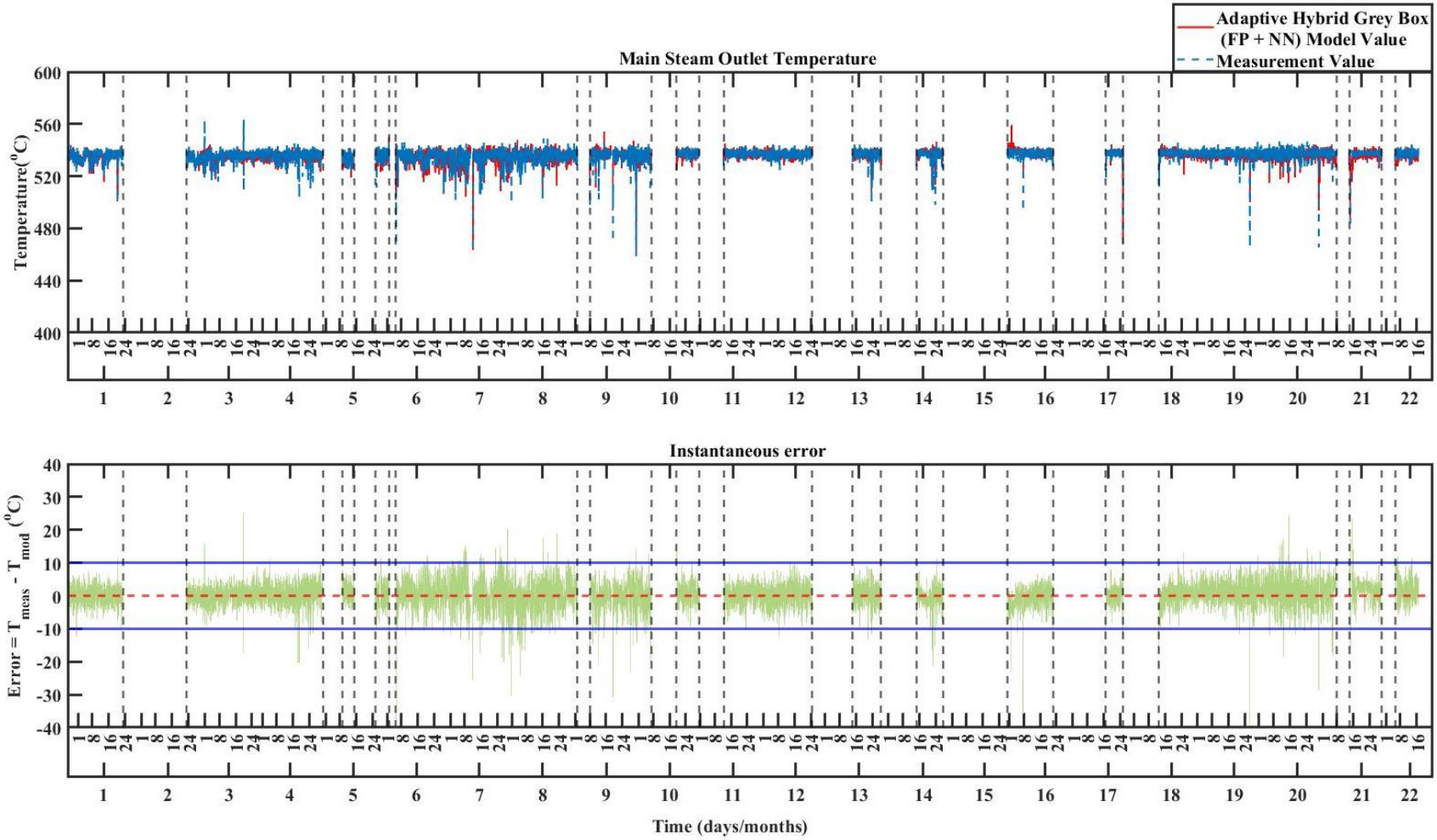


Fig. 5.15. Main steam outlet temperature validation using hybrid grey-box modeling (FP +NN) for 22 months using adaptive training

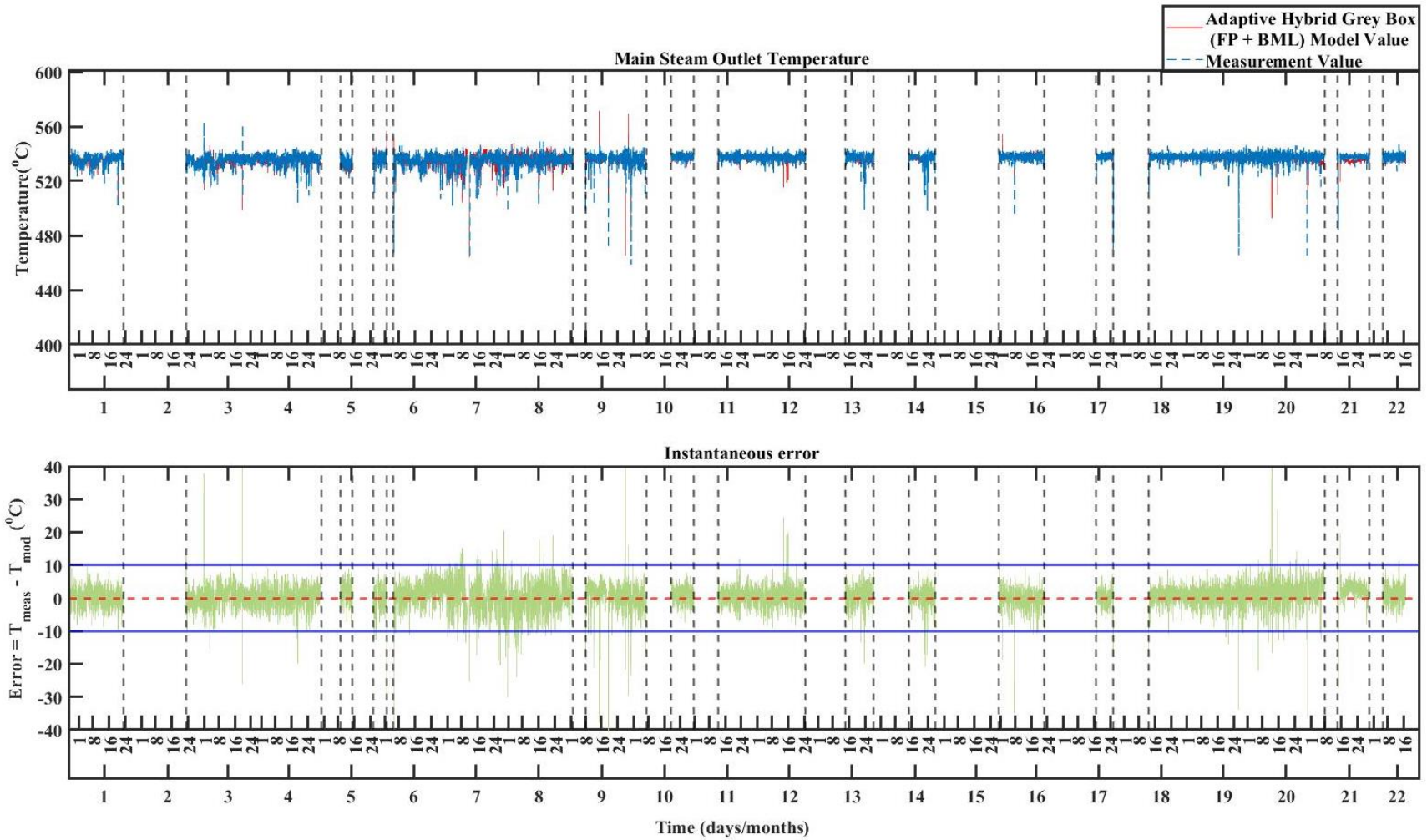


Fig. 5.16. Main steam outlet temperature validation using hybrid grey-box modeling (FP +BML) for 22 months using adaptive training

In both Fig. 5.15 and Fig. 5.16 it is seen that with the use of adaptive training methodology the performance of both FP+NN and FP+ BML have significantly improved as compared to the case when only a 10 days trained base model when used for predicting the 22 months dataset. Moreover, the significant deterioration observed in both models from month 18-2 has improved with the model tending accurately with the measurement data.

5.6.2 Results from Integrated-Parallel Hybrid FP+AI Model

As discussed in Section 5.4, the final superheater system for Plant A under consideration consists of tubular flow circuits stacked one behind another across the heat transfer space. In the actual system configuration, there are 197 flow circuits in Plant A. However, constructing an accurate first-principles model for each flow circuit (location) along the width direction for the given system can be computationally impractical for most cases. Fig. 5.17 shows the temperature distribution in the width direction for the measured data of the steam outlet temperature for Plant A. It is to be mentioned that the legend "Average Measured Data" for the steam outlet temperature, as represented in Fig. 5.17 refers to the measured main steam outlet temperature available directly from plant historian. Additionally, in all the results shown for the integrated parallel hybrid model in Fig. 5.17 and Fig. 5.17, the legends "Max Measured Data" refers to the maximum value (in terms of absolute magnitude) observed and "Min Measured Data" refers to the minimum value (in terms of absolute magnitude) observed for any particular variable among the actual operating data.

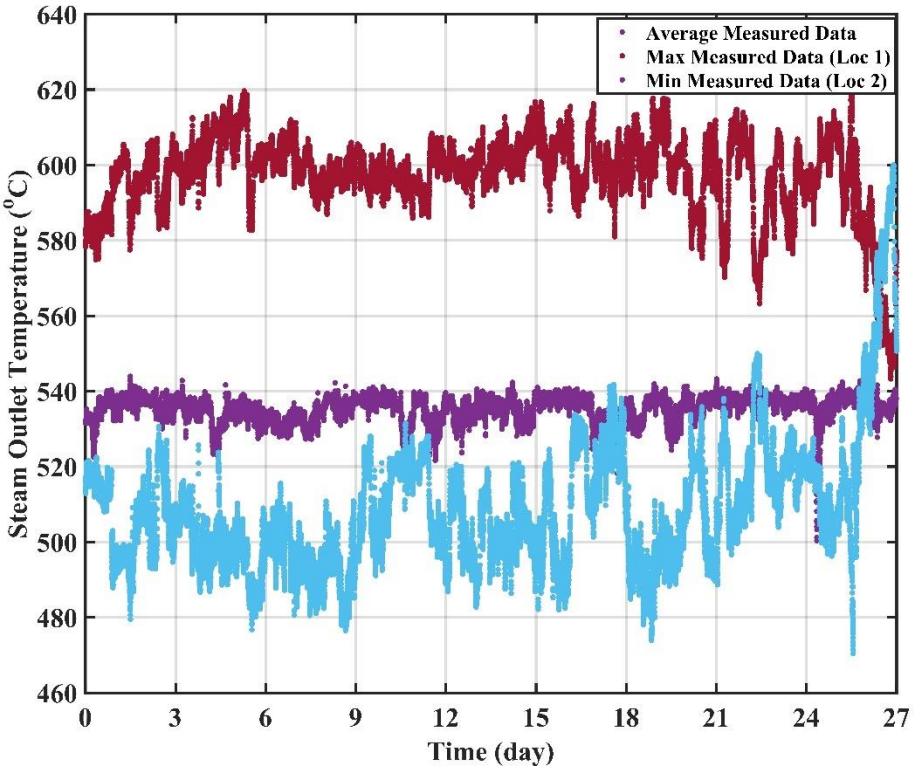


Fig. 5.17. Distribution of transient steam outlet temperature profiles for Plant A, across the flow circuits/locations along the width direction

Fig. 5.17 shows the temperature distribution for the measured data of the steam outlet temperature available from the industrial partner power plant for a continuous operational period of 27 days. In Fig. 5.17, the average measured data (purple dots) are shown, along with the maximum measured data (brown dots) and minimum measured data (cyan dots) representing the wide range of temperature distribution for the steam outlet temperature along the width direction for the superheater in Plant A. In Fig. 5.17,only two datasets for dynamic temperature profiles (termed as 'Loc 1' and 'Loc 2') corresponding to the maximum and minimum deviation from the average measured value, i.e., the steam outlet temperature respectively, have been considered. The standalone FP models developed for the superheater system neglects variability of transport variables in the width direction. Inclusion of another dimension, i.e., the width direction may lead to excessive computational expense and hence, can be difficult for online adaptation. However, accurately predicting the temperature profiles at each tube assembly/location in the width direction will be useful for developing a health monitoring framework for the boiler components for damage prediction in each tube and predict potential tube failures in future. Therefore, such limitations present in a standalone FP model can be compensated with appropriate data-driven models. For such limitations in the FP model, a parallel hybrid configuration of FP and AI models may be beneficial.

In Plant A, the integrated hybrid approach where the AI model is used to represent the correction factor for the heat transfer coefficient is coupled with another AI model in parallel, thus yielding the integrated parallel hybrid FP+AI model. The integrated parallel model developed is then implemented for modeling the dynamic temperature profiles along the width direction of the superheater system in Plants A. Depending on the specific type of parallel hybrid structure, the target dynamic outputs of the second AI models is given by Eq.(5.19):

$$\Delta_{ij}(t) = \boldsymbol{y}_{ij,meas}(t) - \boldsymbol{y}_{i,FPAI_{ser/int}}(t) \tag{5.19}$$

where, Δ_{ij} denotes the deviations, $\boldsymbol{y}_{ij,meas}(t)$ denotes the measured dynamic temperatures at all j tube assemblies/flow circuits along the width direction, and $\boldsymbol{y}_{i,FPAI_{ser/int}}(t)$ denotes each of the respective (integrated) model predicted values of the average temperatures of the i^{th} output variable (refer Fig. 5.5).The model inputs for the second data-driven model, as shown in Fig. 5.5, consists of the inputs to the respective FP model $\left(\boldsymbol{u}_i(t)\right)$, in addition to various other input measurements not used in standalone FP models such as the attemperator spray flowrate, valve openings, and certain inlet steam temperature measurements at specific locations $\left(T_{st,in_j}\right)$ for the integrated parallel model (Plant A). The list of the input/output variables considered for the integrated parallel hybrid FP+AI model can be found in Table 5.3.

Table 5.3 . List of model input and output variables in the integrated-parallel hybrid FP+AI models for the superheater system of Plant B

Variables used in Fig. 5.5	Actual input / output variables for the integrated-parallel hybrid FP+AI model in terms of process variables for Plant B
$\boldsymbol{u_i(t)}$	$T_{st,in}, \dot{m}_{st}, T_{fg,in}, \dot{m}_{fg}$
$\boldsymbol{y_{i,FPAI_{int}}(t)}$	$T_{st,out}$
$\boldsymbol{u_{i,AI_2}(t)}$	$\dot{m}_{spray}, v_{p_A}, v_{p_B}, v_{p_C}, v_{p_D}, T_{st,in_j}$
$\boldsymbol{y_{i,AI_2}(t)}$	Δ_j
$\boldsymbol{y_{i,FPAI_{par}}(t)}$	T_{st,out_j}

The variables involved in the integrated parallel hybrid models represent the same input/output operational variables in the parallel framework, as compared to the individual integrated hybrid FP+AI models and, therefore, not re-defined in Table 5.3. The training data for the AI models in this case are prepared by considering packets of 12-hr operational data. For example, the available data for the entire duration of plant operation under consideration (say, 27 days) are distributed into smaller packets, each spanning a continuous duration of approximately 12 hours based on the steady-state plant load characteristics. Thereafter, around 60% of the total number of those 12-hr packets have been considered for training the data-driven models, followed by a k-fold cross validation step to remove any potential bias in data selection. The predictive performances of the hybrid integrated-parallel FP+AI models have been analyzed while modeling the temporal steam outlet temperatures at individual tube assemblies over a continuous duration of around 27 days, similar to the integrated hybrid models. The steam outlet temperature profiles also portray a considerably higher degree of fluctuations during transients due to variation of plant load frequently cycling among low (450 MW), medium (550 MW) and high (760 MW) load values.

Fig. 5.18 shows the comparison between the integrated parallel hybrid FP+NN and FP+BML models in predicting the steam outlet temperatures distribution along the width direction of the superheater system in Plant A. As before, only two of those locations (tube assemblies) are chosen for demonstration which correspond to the highest and lowest numerical temperature measurements among all tube assemblies. From Fig. 5.18, it can be observed that the hybrid FP+NN model relatively outperforms the hybrid FP+BML model in terms of overall RMSE values and associated computational expense during model training, as shown in Table 5.4. Similar, to the prediction of main steam outlet temperature for the superheater system at Plant A, for this case as well, it is observed that the all-nonlinear parallel (NLS || NLD) static-dynamic NN model performed superior compared to the NLS-NLD or NLD-NLS models and hence, has been chosen to represent the NN model in the hybrid integrated-parallel FP+NN structure.

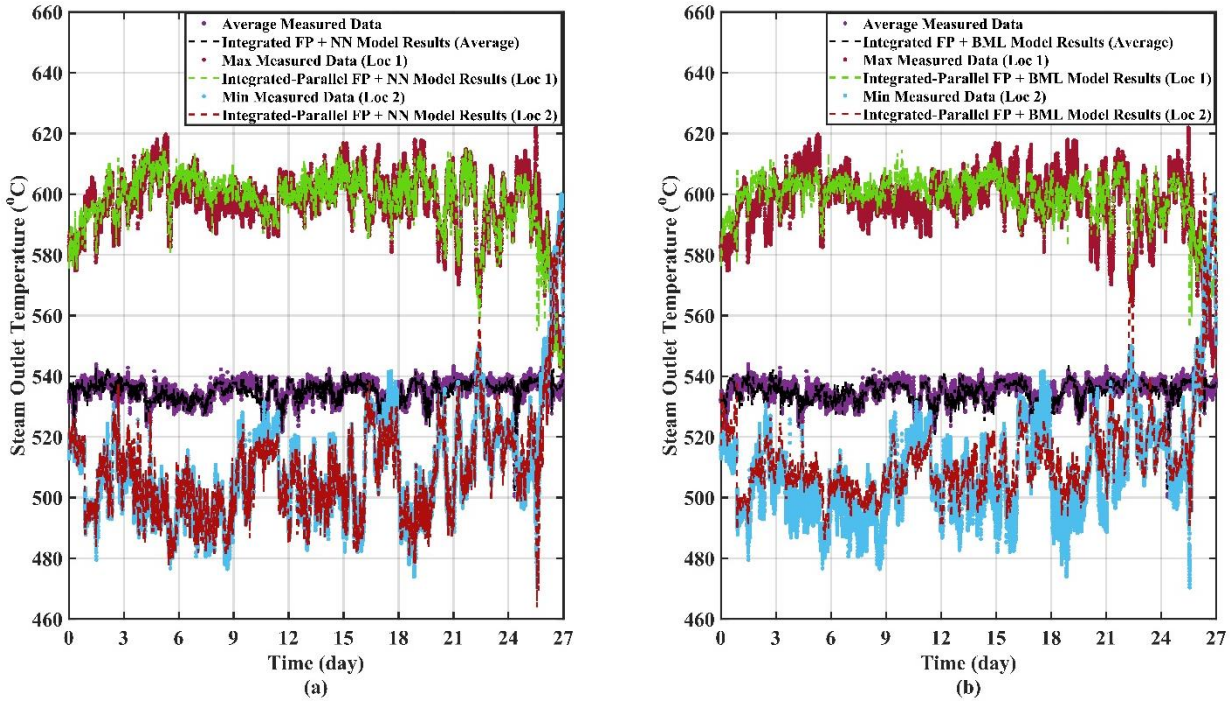


Fig. 5.18. Comparison of results between hybrid integrated-parallel (a) FP+NN and (b) FP+BML models for predicting steam outlet temperatures along width direction with respect to Plant A.

Table 5.4. Comparison between FP+NN and FP+BML models in terms of RMSE, MAIE, and computational expense for predicting steam outlet temperatures in Plant A at individual flow circuits along the width direction

Types of Models	Location 1 (Loc 1)		Location 2 (Loc 2)		Training Computational Time for respective AI models
	RMSE (°C)	MAIE (°C)	RMSE (°C)	MAIE (°C)	
FP + NN	3.1	23.3	3.6	21.5	2 min
FP + BML	6.1	25.7	5.4	27.3	94 min

From Table 5.4, the RMSE values obtained from the hybrid FP+NN model for predicting the steam outlet temperatures at both locations fall within the permissible limits considered in this work. However, the corresponding results obtained from the hybrid FP+BML model slightly exceed the acceptable RMSE value of 5 °C. It should also be noted that the MAIE values observed for this specific case study example are significantly higher than the desired permissible limit defined for this study. Nevertheless, the integrated parallel hybrid FP+AI models successfully enhance (as evident from the RMSE values with respect to measured data, refer Table 5.4) the overall fidelity of the hybrid model by incorporating an additional dimension (variability along width) as compared to the FP model. The higher instantaneous mismatch in the temporal temperature profiles may be attributed to the potential presence of uncertainties in the data under consideration with respect to the coal-fired power plant system (Plant A), stemming from both process noise as well as measurement noise. Furthermore, several other process variables, such as variabilities in coal composition, moisture content, and burner conditions, measurement of which are not available from the industrial partner plant, can significantly impact the temporal temperature profiles.

Furthermore, using the similar adaptive training utilized for the integrated FP+AI models used for predicting the main steam outlet temperature, the integrated parallel FP+AI models is also used to predict the steam outlet temperatures along the width direction for the ‘max’ and ‘min’ measured data for 22 months as shown in Fig. 5.19. As can be seen in Fig. 5.19, the adaptive integrated parallel FP+NN model is able to predict the two locations data with sufficient accuracy with good match observed between the model and the measurement value.

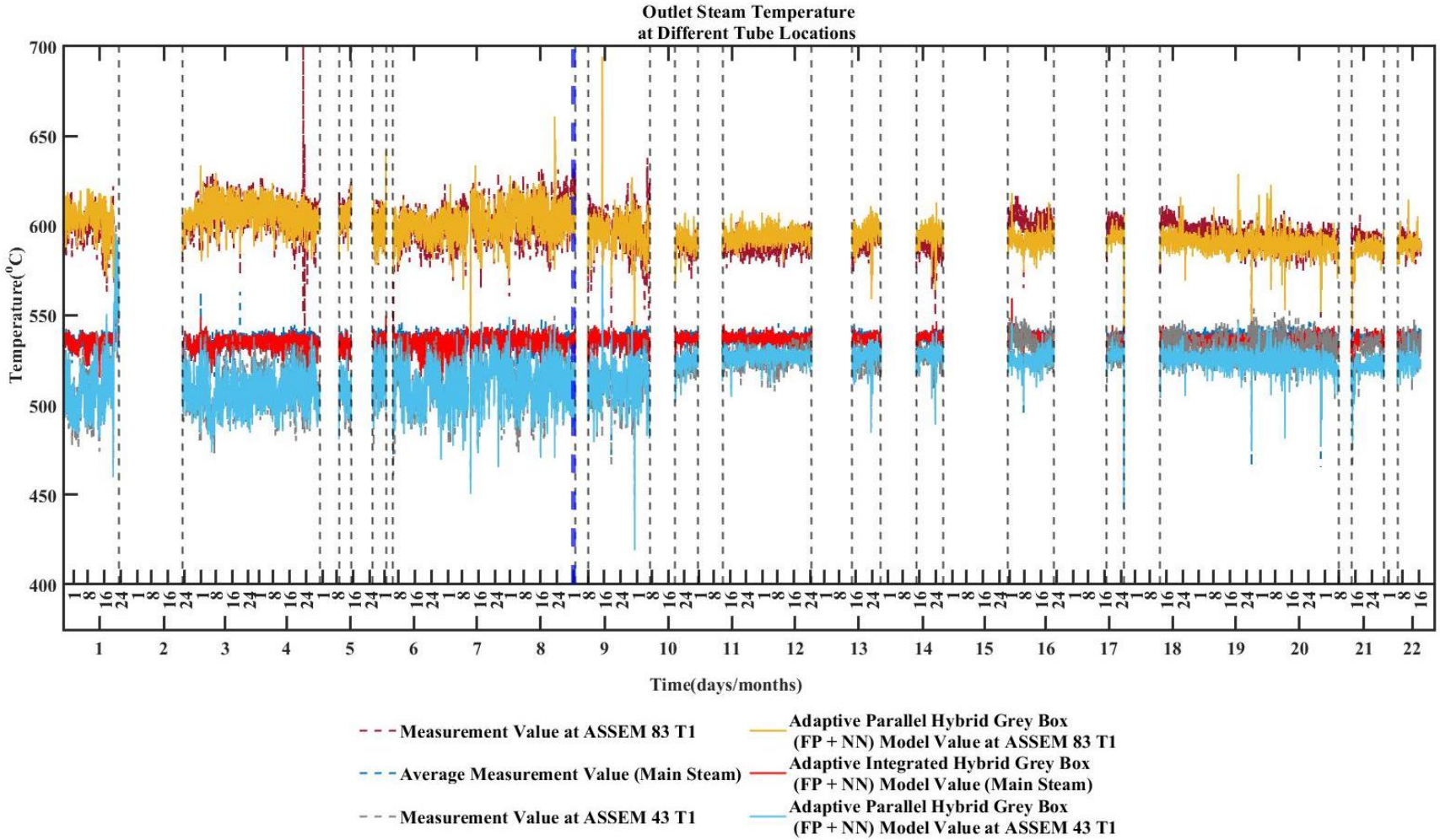


Fig. 5.19. Comparison of results between hybrid integrated-parallel FP+NN models for predicting steam outlet temperatures along width direction with respect to Plant A for 22 months using adaptive training

5.6.3 Results from Oxide Scale Growth Modeling

Adaptive Hybrid modeling of steam-side oxide scale growth

Fig. 5.20 shows the temperature profile of the Dutchmen locations which have been installed in Plant A for a period of 22 months and whose oxide scale characterization has been conducted as described in the section above. The results indicate that that ASSEM 11 T1 has the highest variation in temperature distribution with some larger error observed in the month 18-22 as compared to other dutchmen locations. Also, ASSEM 11 T1 is the location with the highest temperature observed from the dataset and also from Fig. 5.20 it is evident that this location experiences the greatest amount of oxide scale formations among the other locations studied in this work.

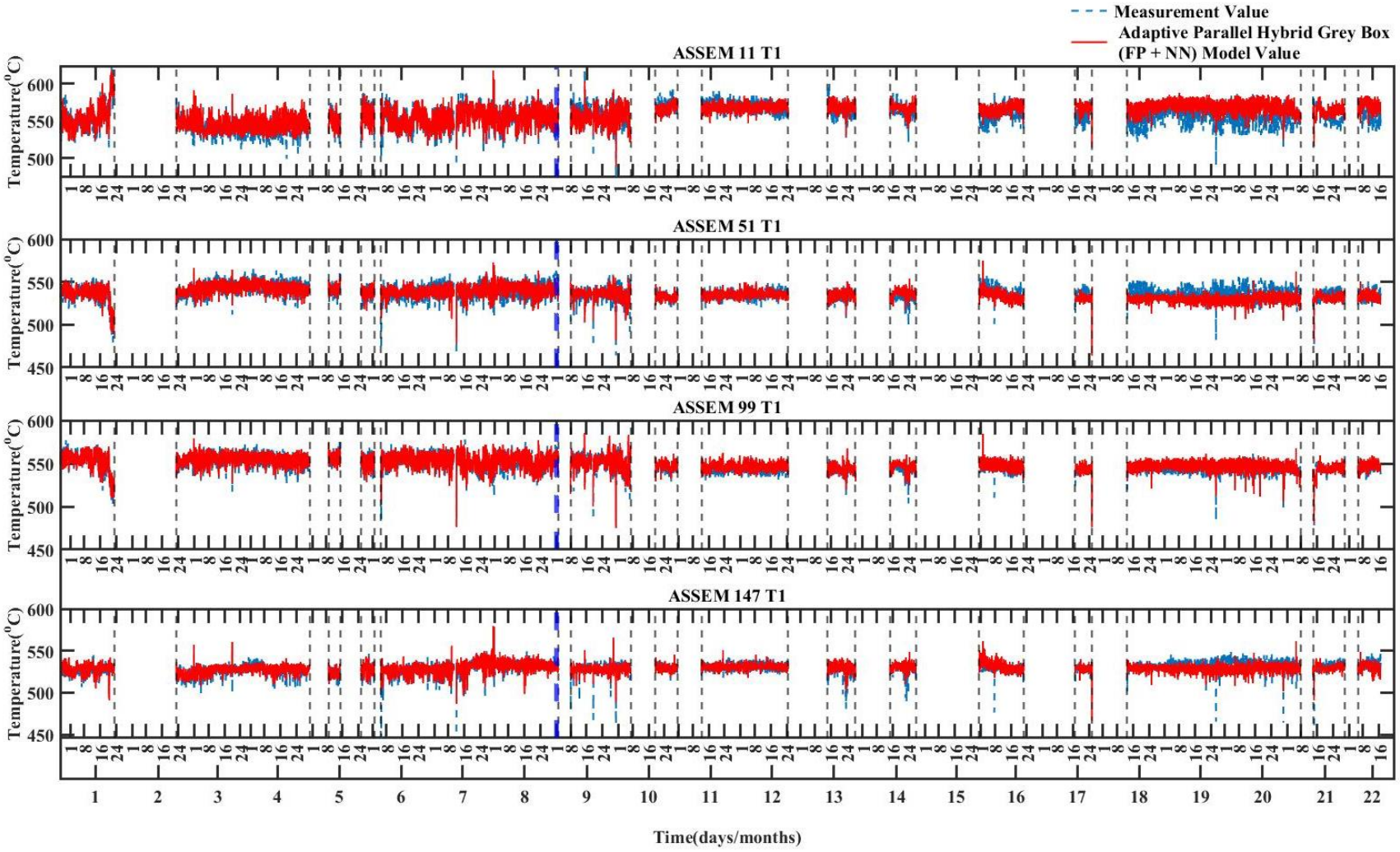


Fig. 5.20. Adaptive Hybrid Integrated Parallel Dutchmen Steam temperatures result for 22 months of data.

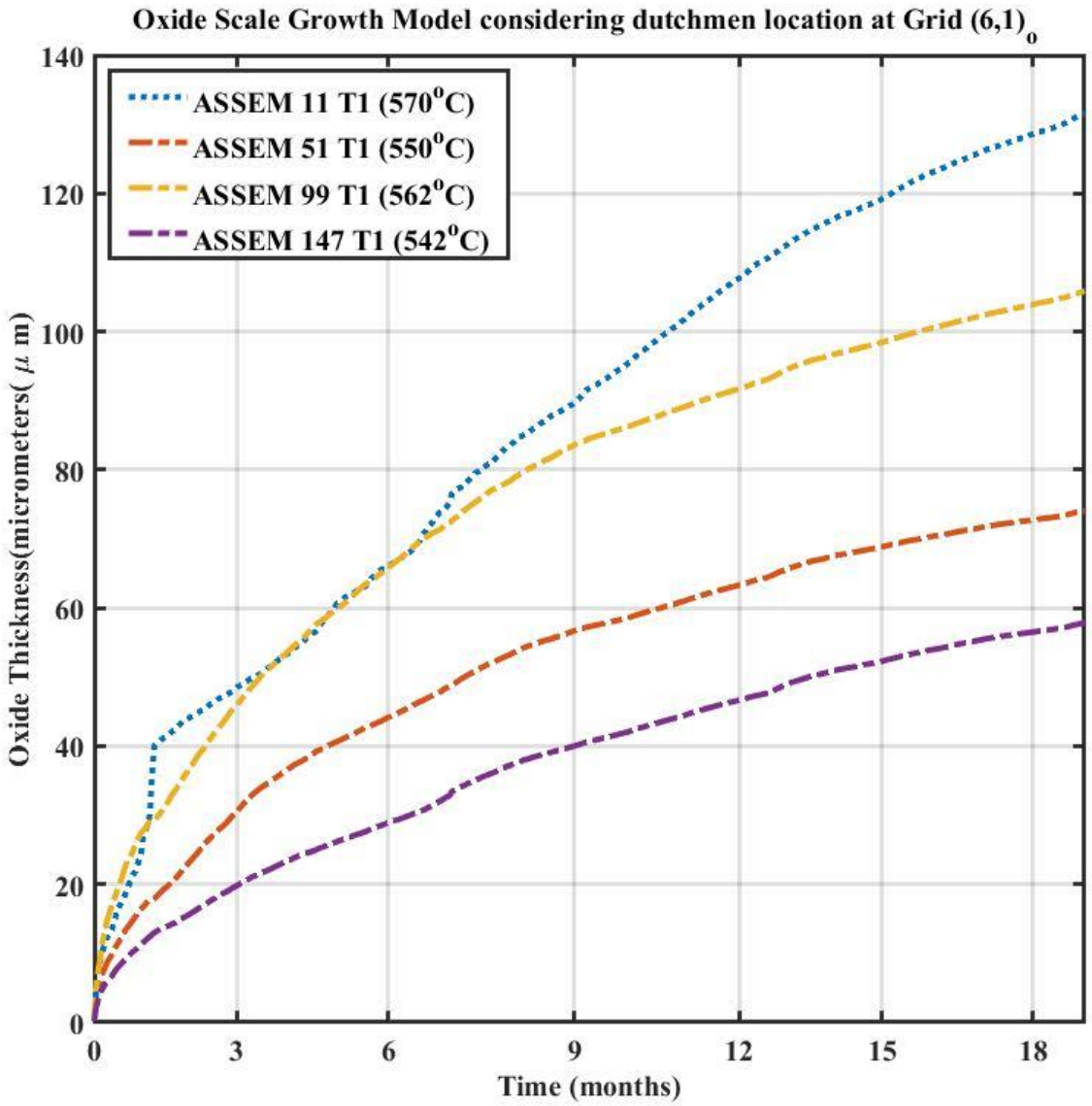


Fig. 5.21. Comparison of Oxide scale growth for the 5 dutchmen locations for the 22 months period

Fig. 5.21 compares the oxide scale growth on various tubes of the final superheater system of Plant A near the steam outlet header for 22 months. As we can see from the figure, the oxide growth corresponding to 4 different tube temperatures (Tube 1) is considered for the system. The temperatures considered in the parabolic growth law are the 22 months dataset, as discussed in the previous paragraph. The average steam temperature for the four tubes is displayed in the legend box in Fig. 5.21 to provide an idea of the expected temperature ranges of each tube. The plot shows that Tube 1 ASSEM 11, which has the highest overall tube temperature of 570° C, has the largest amount of oxide growth, around 135 µm in 22 months. Furthermore, the rate of oxide increase is also faster for Tube 1 ASSEM 11 because of its higher operating temperature. On the other hand, Tube 1 ASSEM 147, having the least temperature of 542° C, will have the smallest amount of oxide growth in 22 months due to its lower operating temperature. The growth rate for all the other tubes is intermediate between Tube 1 ASSEM 11 and Tube 1 ASSEM 147, with most operating near the main steam temperature. Hence, we can see from this analysis that the temperature of operation for each Tube section is the driving force for the increase in the parabolic oxide growth. Thus, this approach is useful for understanding how higher temperatures tubes cause rapid oxide growth relative

to other locations operating at lesser temperatures in the boiler. These places of high-scale growth will be critical to analyze to prevent failure and frequent breakdown.

Table 5.5. Estimates of oxide scale thickness from the adaptive hybrid model and the maximum and minimum values of measured steam-side oxide scale thicknesses of the four tubes.

Tube Number T1	**Model value**	**Experimental Value**	
ASSEM Number	(μm)	Minimum (μm)	Maximum (μm)
11 (Right)	131.6	129.3	172.7
51 (CR)	74.08	79.8	100.0
99 (Center)	105.8	98.0	119.2
147 (CL)	57.84	31.3	97.0

Table 5.5 shows the model predicted oxide scale values and compared it with the maximum and minimum values observed form the experimental analysis .It can be seen from the Table 5.5 that ASSEM 11 has the highest amount of oxide scale predicted by the model and ASSEM 147 has the least which matches the exact ranking order as observed from the experimental data. Although, the model absolute values for ASSEM 51 is not within the range of max and min and ASSEM 11 model value is very near to minimum value. However, the ranking of the model values matches exactly with the experimental data and the deviation in the absolute model values can to due to significant uncertainty in measuring the exact oxide scale thickness from the tube specimens considered.

5.7 Conclusions

In summary, the proposed hybrid structures are found to yield accurate predictive capability and enhanced interpretability while modeling the transient temperature profiles of the superheater system at the partner power plant (Plant A) under consideration, when tested with actual industrial data. In all the different case studies considered in this work, it has been observed that the hybrid FP+AI models significantly outperform the standalone FP models, thus improving the overall performance of the hybrid models. Furthermore, it has also been observed that in all cases, the hybrid FP+NN models are associated with significantly lesser computational expense during model training as compared to the hybrid FP+BML models. However, it is considerably difficult (if not impossible) to generalize the superiority of one type of hybrid FP+AI model over another because the performance of a specific type of hybrid model in achieving a particular objective depends on the respective advantages and disadvantages of the associated AI models in the hybrid architectures. Finally, we would like to note that it is considerably difficult to

arrive at a universal conclusion on the superiority of a particular type of hybrid FP+AI model over another since the efficiency of such synergistic integration of FP and AI models largely depend on several factors such as the physics/chemistry of the specific system, type of information transferred between the models, characteristics of data available for training the data-driven models, and candidate architectures/algorithms associated with the development of the hybrid framework. In general, it is our opinion that for modeling complex dynamic systems, various structures of the hybrid FP+AI models can be exploited without necessarily developing a very high-fidelity FP model that may require large resources and can incur excessive computational cost. Although, the fidelity of the FP model and the capability/characteristics of the AI models depend on specific system and the variables to be represented by the hybrid models. We believe that the proposed hybrid FP+AI structures and AI models exhibit sufficient flexibilities to be implemented independently or in combination with one another, not only for modeling industrial superheater systems, but also many other complex dynamic nonlinear systems.

6 Validation and Integration of Hybrid Model at Plant A

6.1 Overview on Boiler Health Management

Power boilers are complicated because they seek to ensure effective combustion of pulverized coal and then the efficient capture of the heat produced. The general arrangements for this process in commercial plants typically attempt to ensure that a uniform flame is produced at each of a number of burners. The fire ball produced is typically surrounded by a ‘water wall’ of steel tubes which extract heat and boiler water which is circulated through them. In many boilers the heat captured by the ‘water walls’ will result in boiling of the water inside. Then a steam separation system typically extracts the steam, and this is introduced to a superheater section of tubes. The superheater is important because the associated tubes increase the temperature and pressure of the steam increasing the thermal efficiency of the electricity produced.

It should be emphasized that a power boiler used for a unit generating say 500 MW of electricity will have a boiler with miles of tubes which may be manufactured from as many as 10 different alloy steels. Design is typically carried out based on the ASME Boiler code and the expertise of the individual manufacturer. In general, proper material selection gives the opportunity for the least cost steels to be used in different locations without compromising performance and reliability.

After new construction of a boiler, standard practice is to perform a hydrostatic test on the boiler. This involves imposing a pressure to the tubing system as a straightforward approach to find leaks; this pressure may be above that of the designed operating condition. Once the initial quality assurance has been completed operation will continue to fulfill the demands for electricity from the end users. Historically, most efficient coal fired boilers operated at or close to rated capacity in a ‘steady’ or ‘base load’ condition. More recently, demands for electricity from coal fired boilers have become more varied so these units now frequently change unit load and are also used as back up, i.e., only generating in periods of peak demand.

The water and steam systems in an individual boiler are designed to operate such that steam is delivered to the turbines at the desired flow volume, pressure, and temperature. The water and steam systems are interconnected and generally tube failures will require a boiler to be shut down for repair. Most power generating companies impose target levels of availability for specific boilers in their system. These targets are a key factor in ensuring that reliable power is available to the electricity system when the load requires it.

Experience has shown that boiler tube failures (BTF) are traditionally the premier cause of forced outages of coal fired power generating units worldwide. Historically, costs for BTF in the U.S. are estimated to be more than $1B/yr in power replacement charges, repair, and maintenance, combined. In an individual

power station, one BTF typically results in a forced outage lasting three days and can cost >$3 million for replacement power. EPRI has worked with manufacturers, end users, and other stakeholders to establish cost effective approaches to ensure that boiler reliability meets expectations. These interactions have shown that over-temperature operation is a significant cause for BTF. Locally elevated temperatures within tubes can be a consequence of factors such as excessive heat in the combustion zone, non-uniformity of the heat patterns within the boiler, and changes to the flow of water or steam. When the elevated temperatures are sustained over time, damage develops by a mechanism known as creep (time-dependent damage development). The rate of creep damage is directly related to the temperature and pressure in the tube, as well as the alloy used for tube manufacture. Typically, ASME design rules will specify allowable stress values – which are calculated from tube dimensions and the internal steam pressure – for specific temperatures. These allowable stress values are specified for individual alloys which are used in the waterwalls and superheater / reheater of power boilers. Traditionally, individual boiler manufacturers would use company guidelines (which may include ASME design rules) to ensure that the tubes selected were not required to operate at temperatures which would result in rapid deterioration in properties.

Generally, when units were operated under base load conditions, superheater and reheater tube life was governed by creep properties and the influence of external tube corrosion and erosion from combustion products which resulted in wall loss. The level of damage present would typically be evaluated by performing manual inspections during outages when the boilers were under ambient conditions, suitable for human occupation. In general, damage accumulated over time is a consequence of high temperature creep and wall thinning due to erosion or corrosion. More recently, coal fired power plants have been operating in cyclic mode as their priority in system dispatch becomes lower. In other words, in most generating systems, nuclear and renewable generation will be called to fulfill system load demand first and fossil units will operate in a variable output mode to compensate for system load fluctuations, or even in a back-up or reserve capacity. In many cases, these power plant operating regimes are significantly different than those envisioned at the time of design. These variable generating load conditions introduce uncertainties regarding the local operating conditions within the boiler. It is an incorrect assumption that because the boiler is generating less steam and hence less electricity that the temperatures and pressures within individual tubes will reduce.

Given the expected operating conditions of reheater and superheater tubing, an oxide scale can be expected to develop on the steam side of the tube. The scale generally grows according to a law where the thickness is a parabolic relationship with time where, as time increases, the rate of growth will decrease. This should be emphasized because as this steamside scale increases in thickness, the rate of heat transfer from the metal tube to the steam will decrease because of both the increasing oxide layer thickness and the inherent low thermal conductivity property of the oxide scale material. This change will have the effect of increasing the metal temperature which in turn increases the rate of scale growth and so on. Thus, it is apparent that as scale increases in thickness the tube temperature will increase. This will accelerate the rate of creep damage development. A simple estimate indicates that for a temperature increase of 25C (45F) creep life is reduced by a factor of approximately six. Life reductions of this level are significant and will have a dramatic effect on boiler reliability, maintenance, and repair.

6.2 Demonstration at Plant A

A principal pillar of this project was the direct comparison of model predictions leveraging real-time data to material exposed in the plant over the same duration. The project team worked in collaboration with Southern Company and one particular coal fired boiler in their fleet, here designated as Plant A. The team fabricated five lengths of tube, instrumented them with their own thermocouples, installed them in the final superheater in 2021, returned the plant to service on-schedule, and extracted the tubes after approximately 10,000 fired hours. This section describes the selection process, implementation of the installed tubes, and data analysis. Section 6.3 discusses the ex-service metallographic study and the basis for comparison to the model predictions.

6.2.1 Value of a live demonstration

This project is aimed at not only constructing a hybrid physics-based and machine learning model of the real-time temperature distribution throughout a boiler, but also the physical validation of the inputs and predictions of that model through collaboration with a power plant. The work scope involves predicting the evolution of damage as a function of local temperature in both space and time. Integral to that prediction is validation through quantified measurements via sensors and periodic metallurgical inspection. The installation of hardware in an operating power plant is a critical component of achieving these goals.

Due to the complex nature of the proposed hybrid model, the project team wanted to pursue validation of the model outputs with witness material. This need to validate the model was echoed by industry experts at each presentation made by the project team. Using live data from an active coal fired power plant, as it responds to demand and the needs of the grid, covers the project needs to incorporate the impact of flexible operation on local temperature distributions in the final superheater.

The data collected from the unit during the project, covering approximately two years of operation, served as a model for typical operation of a large coal fired boiler. This data served as an input to the hybrid model to understand the breadth of operational flexibility it needs to accommodate temperatures, heating and cooling rates, cyclic frequency and periodicity, etc., as well as component specific data relative to various header geometries and locations.

6.2.2 Reason for selecting this plant/unit

The unit involved in this study was a 700MW coal fired power plant burning Colombian sub-bituminous coal, Plant A. It was designed for a steam outlet temperature of 537°C (1000°F) at 24 MPa (3,500 psig), Fig. 6.1 and features a superheat and reheat sections that both consist of pendant assemblies.

The project focused on a technical demonstration in partnership with Plant A. Partnering with Plant A offered a substantial opportunity to satisfy the goals above but presented challenges as well. For example, the project was required to align within the outage schedule of the plant and take advantage of the downtime that afforded for insertion and extraction of tubes. Microstructural damage and oxide growth take time to develop and having a longer window between inspections means that much more change can be observed; in fact, having too short a window risks insufficient oxide growth to accurately measure.

This unit was selected because it provided unique opportunities for the project. Specifically, this unit was expected to operate sufficiently during the project duration, ensuring accumulation of fired-hours on test specimen. This unit also had a high degree of instrumentation that preexisted the project, associated with the tubing and headers. This included 75 thermocouples on the tubes leading into the final superheater inlet and outlet headers.

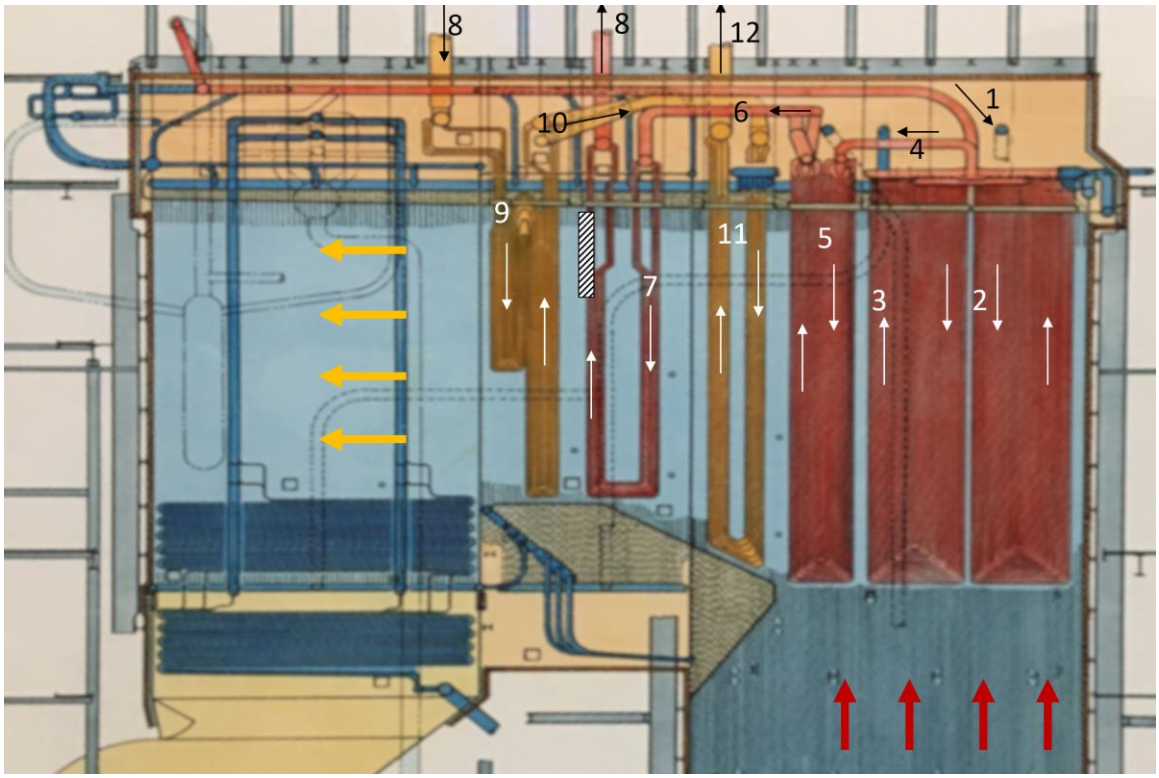


Fig. 6.1. Schematic diagram of the configuration of tube banks in the Plant A boiler including the location of the added dutchmen as a hatched box in the final superheater (7). White arrows show the direction of steam flow. Red and yellow arrows show the flow of flue gas.

The model developed by WVU aimed to predict temperature distributions throughout a portion of the superheater. The configuration of tubing in the superheater in Plant A consists of 197 assemblies of 6 tubes; a total of 1182 tubes instrumented in the penthouse with 17 inlet thermocouples and 58 outlet thermocouples. The average superheater tube in this boiler is approximately 25 m in length and the superheater alone represents over 55 km of tubing. Steam enters the superheater at approximately 500°C (930°F) and exits at 540°C (1000°F).

The portion of the superheater circuit in the unit targeted for the subject model of this study consisted of 347H stainless steel tubing. This alloy represented a challenge to the project. Specifically, the project relied on the steam side oxidation as a key marker in its modeling efforts. Stainless steels oxide growth rates are approximately 5% as fast as ferritic low-alloy steels, such as SA213-T22 (2.25Cr-1Mo). For the project to measure significant change (oxide growth) over the span of the experiment (2 years), the project had to introduce selected lengths of T22 tubing into the superheater tubing circuit. This was acceptable to Plant A and the project team because of extensive industry experience with and knowledge of this alloy. Globally, T22 is one of the most prevalent superheater tubing alloys in coal burning boilers.

6.2.3 Location in the Superheater

The demonstration aspects of the project were focused on the pendant spaced superheater, also known as the final superheater, described above. This was selected as the primary component of interest for several reasons. The final sections in both the primary heat and reheat lines experience the highest metal temperatures which would be expected to accelerate thermally activated damage mechanisms in the tubes. Specifically, for a given alloy, oxide growth rates are primarily related to time and temperature, so the maximum tube metal temperatures are desired for observing the most significant metallurgical change over the limited operating window for the project duration. The superheater is preferred over the reheater due to the elevated pressures in the former; higher pressures require thicker tube walls which gives the experiment greater temperature gradients through the tube wall.

While all options bring technical and logistical challenges, the plan for sensor installation was focused on the pendant spaced superheater in Plant A. In 2021, Plant A installed ASME code-compliant tube lengths, also called dutchmen, in the final superheater with attached surface thermocouples. These segments were each 2.4 m (8 feet) in length and were fabricated using T22 tubing with 347H tubing safe ends. These segments were located at the very end of the final superheater, near the penthouse and tube outlet, as indicated by the hatched block in Fig. 6.1.Five dutchmen were located at assemblies 11, 51, 99, 147, and 189. Fabrication, inspection, and installation of these dutchmen was performed by subcontractors to Plant A – companies with which Plant A already has experience – in order to streamline technical communication, logistical planning, and risk related to operating the commercial boiler.

6.2.4 Dutchmen Design and Fabrication

The tube lengths installed in Plant A were fabricated by Southern Company contractor Boiler Tube Company of America (BTA) of Lyman, SC. These dutchmen were comprised of the following subcomponents, also shown in Fig. 6.2.

- Two 305 mm lengths of 347H tubing, 44 mm OD x 6.6 mm wall thickness, which acted as safe ends for on-site welding.
- One 1320 mm length of T22 tubing, 44 mm OD x 10.5 mm wall thickness, to-which were attached two thermocouples
- Two type Type K thermocouples attached at the 2-foot (610 mm) and 6-foot (1830 mm) elevations to the T22 tube

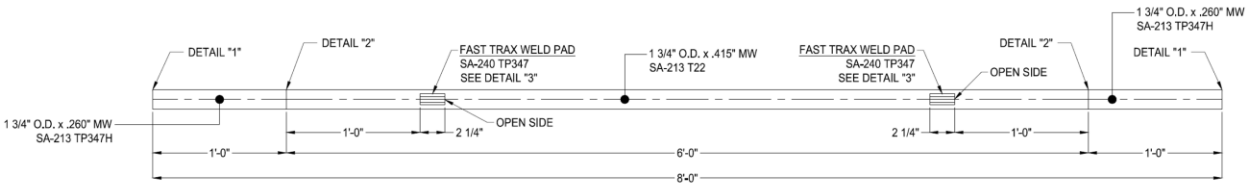


Fig. 6.2. Excerpt of the BTA drawing for T22 dutchmen to be installed in Plant A. Dimensions in inches.

To match inner diameter of the T22, the 347H tubing was swaged to an ID of 23 mm and then heat treated. All fabrication welding was performed to BTA's standard practices using SCS-approved procedures. This ensured a level of quality required by Plant A prior to installation into the superheater section of the boiler. All of the welds necessary to fabricate the dutchmen passed radiographic testing.

Five such dutchmen were fabricated for installation in Plant A and a sixth was shipped to EPRI for evaluation of the as-received condition. This evaluation helped characterize the starting condition of the tubing which was a valuable input to initializing the WVU physics-based model.

6.2.5 Installation

Installation of these dutchmen and associated thermocouples was successfully completed in 2021. The plant returned to service unhindered by the work associated with this project. The dutchmen were extracted in 2023 after an accumulation of 10,000 hours of high temperature operation.

6.2.6 Analysis of Operational Data

Prior to the installation of these dutchmen, the project team discussed the details of where to locate the dutchmen in the superheater: does the installation emphasize one region, one half, or the entire boiler width? To facilitate this decision, the project analyzed operational data from the final superheater to examine the distribution of temperatures across the width of the superheater. From an approximately 1-month operational window from 2021, the temperatures were observed to be significantly higher near the walls of the boiler, as shown in Fig. 6.3. This figure plots the inlet temperature, outlet temperature, and minute-to-minute average temperature at the four locations in the superheater where inlet and outlet thermocouples are available. Of the 198 tube assemblies across the width of the superheater, these TCs were located at the inlet and outlet of assemblies 3, 51, 99, 147, and 195; however, outlet temperatures were not reported on tube 99. The higher temperatures observed on the edge regions of the boiler encouraged the project to monitor those locations as higher temperatures would be expected to drive faster damage evolution.

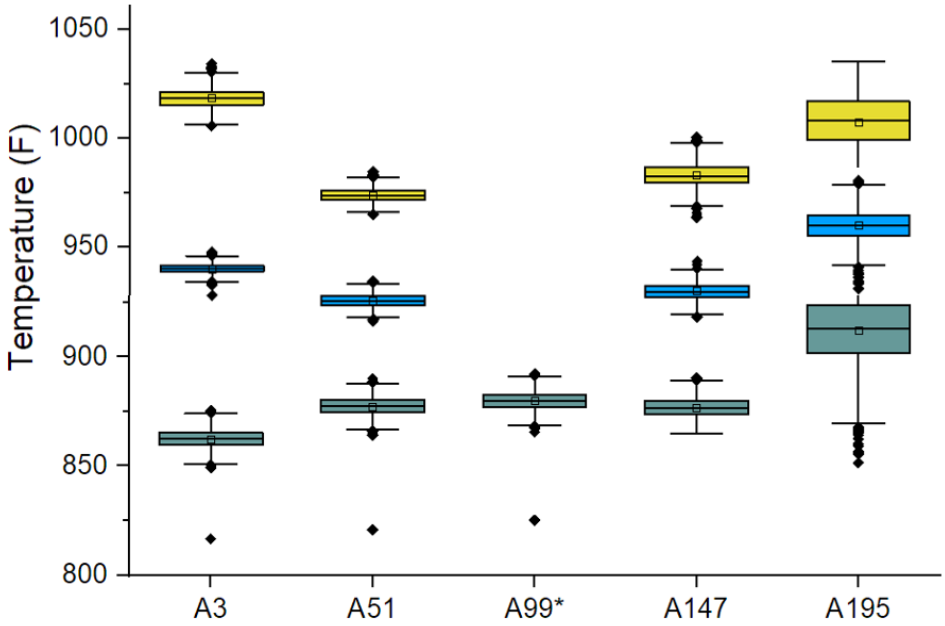


Fig. 6.3. Boxplots of temperature distribution across the width of the final superheater over a 1-month timespan. Green is inlet temperatures, blue is the mean temperatures, and yellow is the outlet temperatures.

The project team evaluated the thermocouple data provided from the superheater in Plant A over the span of the experiment to examine it for trends and risks. This effort represents a baseline for what sort of analysis plant maintenance staff could perform with the information available today. The WVU hybrid model predictions were expected to provide information above and beyond this current-state analysis. For example, the current analysis is backward-looking only; any forward predictions are merely based on the assumption that the plant operates as it has. Further, the analysis of thermocouple data is only possible at the locations where the TCs are physically installed: these are external metal temperatures, not the internal steam temperatures in the penthouse, and only on 58 tubes of the total 1,182 tubes in the superheater which have installed TCs. In reality, the resolution and number of locations providing data were lower due to TC failures: in most data sets provided for the superheater, the analysis below considered TC information from about 45 locations.

(It should be emphasized that the number of TCs installed in the superheater in Plant A is outstandingly many and is among the most in the industry. Most boilers and HRSGs have only a few TCs to monitor tube temperatures.)

The hybrid physics and machine learning model should be able to overcome the limitations of having to rely only on physical measurements by using heat flux calculations to estimate temperature throughout the length of the tube and at more locations throughout the superheater, filling in the gaps between existing TC measurements.

Analysis of the TC data provided consisted of two primary monthly metrics: the peak observed steam temperature and a semi-quantitative assessment of aggregated temperature exposure. A monthly peak temperature is easy to report and provides an easy, if incomplete, screening for risk. The aggregated temperature exposure requires some clarification: this aggregated temperature exposure calculation uses the time-temperature relationship for creep damage accumulation for T22 as an example of how the assessment could be performed. For example, temperature increases of 28C (50F) typically result in six times faster accumulation of creep damage. This is in contrast to the peak temperature analysis which, alone, may be overly sensitive to a short duration temperature aberration.

The aggregate exposure analysis was performed for each minute of reported TC temperature at all locations. The damage fraction for a given minute is dependent on the instantaneous temperature. At any given temperature, T, in Rankine, the estimated life can be calculated from a Larson-Miller expression as follows:

$$Life_T = 10^{\left(\frac{LMP}{T} - C\right)} \tag{6.1}$$

and

$$LMP = A + B * UTS + D * \log(\sigma) + E * \log(\sigma^2) \tag{6.2}$$

This assessment for average T22 material used the coefficients: A = 40,975; B = 57 and a UTS of 70 ksi; C = 20; D = -5,225; and E = -2,450[396].

The accumulated damage, d, for a given set of exposure times (e.g., 1-minute intervals), t, is calculated as:

$$\boldsymbol{d = \sum_{t} \frac{t_T}{Life_T}} \tag{6.3}$$

This assessment assumed constant stress, 5800 psi (40 MPa), implying uniform tube diameters and constant steam pressures. This is expected to be not strictly true, but the fluctuations in stress are assumed to be second order compared to the influence of temperature. Again, this is a demonstrative assessment; a proper life assessment could consider much more plant-specific inspection data. Tweaking this analysis to use lower-bound material properties or to assume–or calculate–a gradient from steam temperature to metal temperature could be useful in increasing the magnitude of calculated damage; qualitatively the highest risk areas could be expected to remain the same regardless of assessment. Note that lower bound properties are typically 80% of the mean in stress or approximately 25-30% of the mean in life.

Fig. 6.4 shows the peak temperature values reported at each outlet TC location on a month-to-month analysis March 2022 through February 2023. The highest temperatures are generally above the manufacturers recommended operating conditions which would be expected to yield shorter useful life. Tube assembly number increases from the left side of the boiler to the right and the vertical rows are the

rows of each assembly (1 to 6). Consistently, A83-R6 shows some of the hottest temperatures.

Fig. 6.5 shows the aggregated temperature exposure values as calculated above, again for March 2022 through January 2023 (the plant did not run for much of February 2023). Here the color scheme is absolute, but qualitative for the reasons already discussed. The intent is not to provide quantitative values of damage but the presentation method does show regions' relative damage accumulation. Although the right side of the boiler superheater shows higher peak temperatures, few tubes there are at as significant risk from long-term overheat damage as those in the center.

This type of data analysis is something the plant operator could perform with limited effort to provide a screening of boiler operation on a month-to-month basis. It could be helpful in identifying operational issues like blockages or maldistribution of gasses or it could provide one piece of a maintenance plan towards further inspection. In this example data set, tubes assemblies A3, A11, A59, A67, and A179 appear to have accumulated the most risk and would benefit from further inspection if that trend continues.

The project team completed a second assessment, this one of the full operating data (23 months) for the five dutchmen locations and corresponding penthouse TC locations. The goal for this data analysis was to evaluate different measures of exposure temperature to assess the impact operation had on the tubes. The metallographic evaluation, particularly for oxide scale thickness (Section 6.3) represents a snapshot of the end-state. Different data analysis methods offer other perspectives which should line up with the physical evaluation or provide additional directional indicators.
The thermocouple data was analyzed from each of the key thermocouple (TC) locations: in the penthouse, those locations were tube assembly A11 tube T6, A51T6, A99T6, A147T6, and A195T6. These correspond to the dutchmen locations on the outer most tube (T6) at assemblies 11, 51, 99, 147, and 189. Unfortunately, the TC location A187T6 which would be closer to the actual tube location of the dutchmen (A189) did not report any data; A195 was substituted. Histograms of thermocouple data for the exposure times for each location is presented in Fig. 6.6.

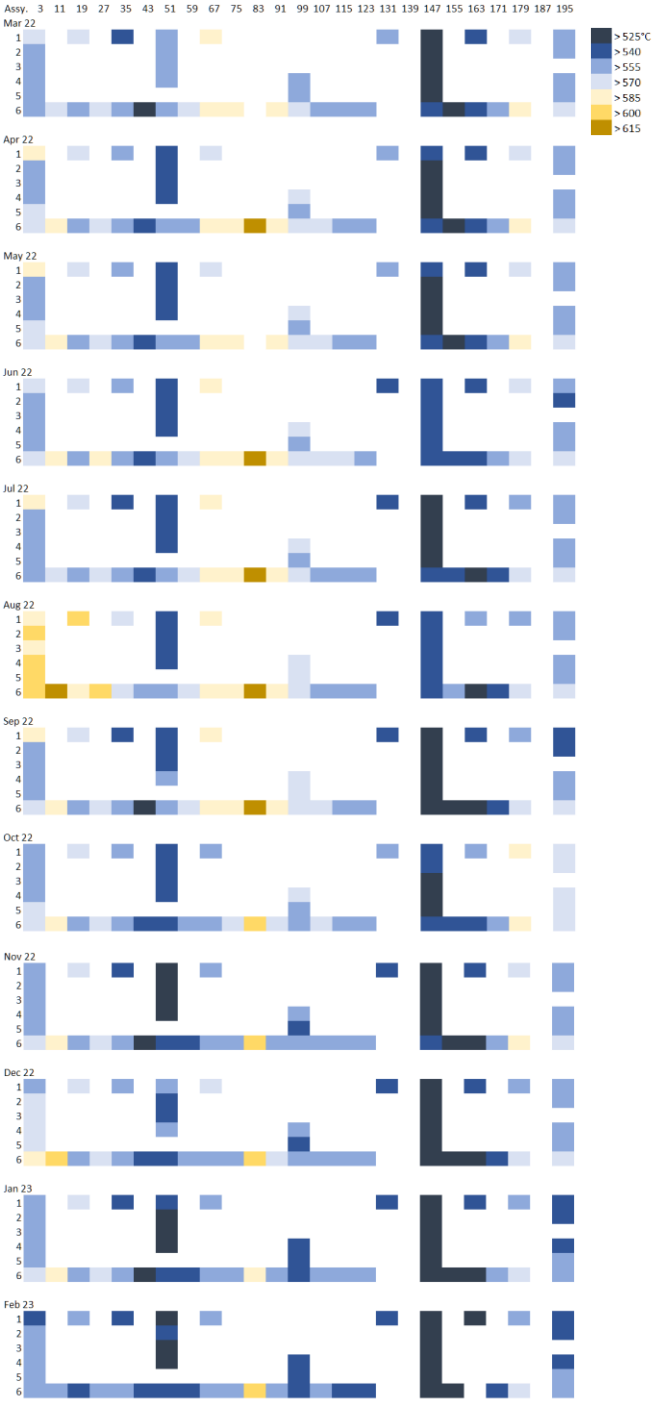
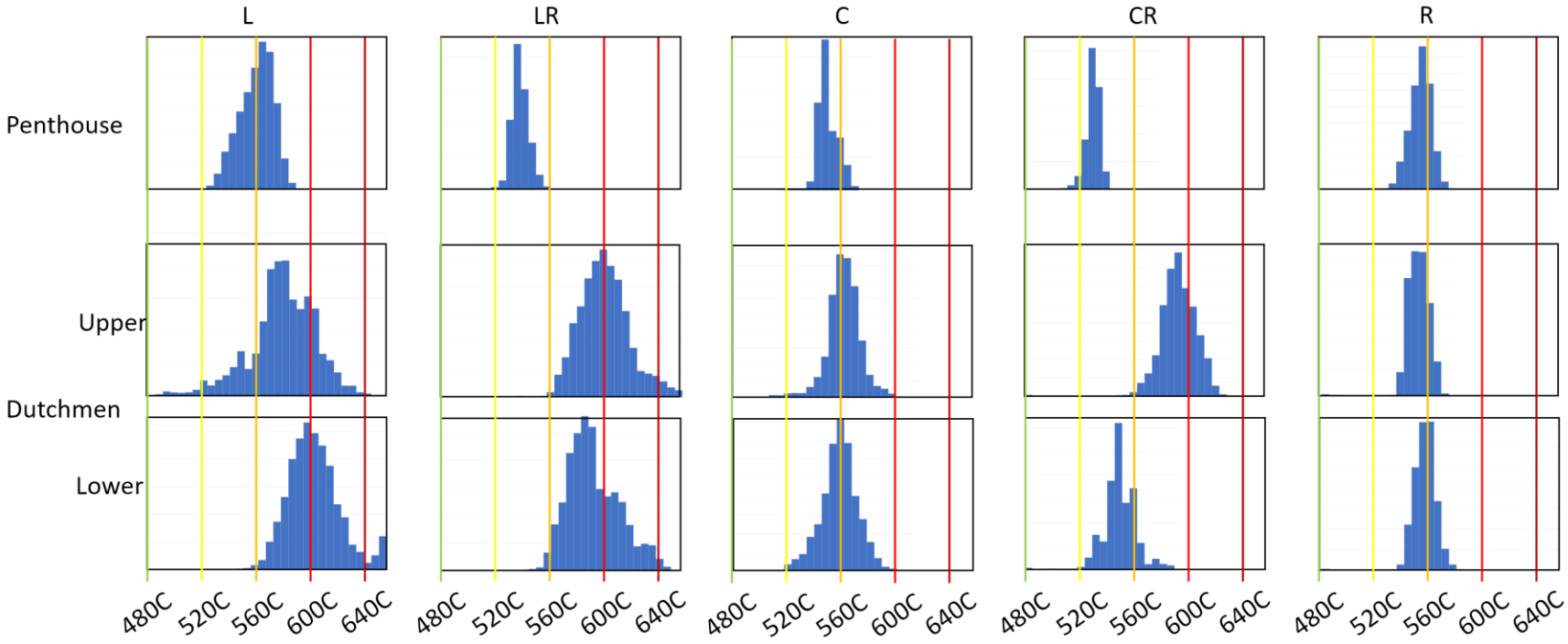


Fig. 6.4. Peak temperatures reported for TC locations in Plant A superheater for each month March 2022 through January 2023. The colors range from dark grey (525°C) to dark yellow (615°C) and white is no data or below 525°C.

Fig. 6.5. An approximation of aggregated damage for TC locations in Plant A superheater for each month March 2022 through January 2023. The color scheme is a relative index of damage, not quantitative. Dark blue indicates the least damage, dark yellow the most. White indicates no data.

Fig. 6.6. Collection of normalized histogram plots showing the thermocouple data at the locations of interest.



6.2.7 Flue Gas Survey at Plant A

The heat transfer model developed by WVU for Plant A requires detailed knowledge of the flue gas temperature and how the temperatures and flow conditions vary at different locations and how they fluctuate with operating conditions. While estimates of typical flue gas conditions exist, this project had an opportunity to directly measure temperatures during operation. These measurements provided a much more realistic picture of the influence of reduced load and transient behavior on flue gas conditions ahead of the final superheater. This linking of load to flue gas temperature and flow is crucial for producing a "signature" of the boiler, discussed in more detail below.

The project team contracted with a vendor specializing in boiler evaluation for measuring flue gas temperature. High velocity thermocouple (HVT) systems are an industry standard approach to not only temperature measurement in extreme conditions but also velocity and gas constituent sampling. The system used in Plant A's boiler extended a 20-foot long, water-cooled lance into the boiler (Fig. 6.7). Given a boiler is about 200 ft wide, with ports on both sides, a 20-ft lance accounts for 20% of the boiler width. At the tip was a protected thermocouple, generally K-type (effective up to 1100°C), as well as a gas collector.

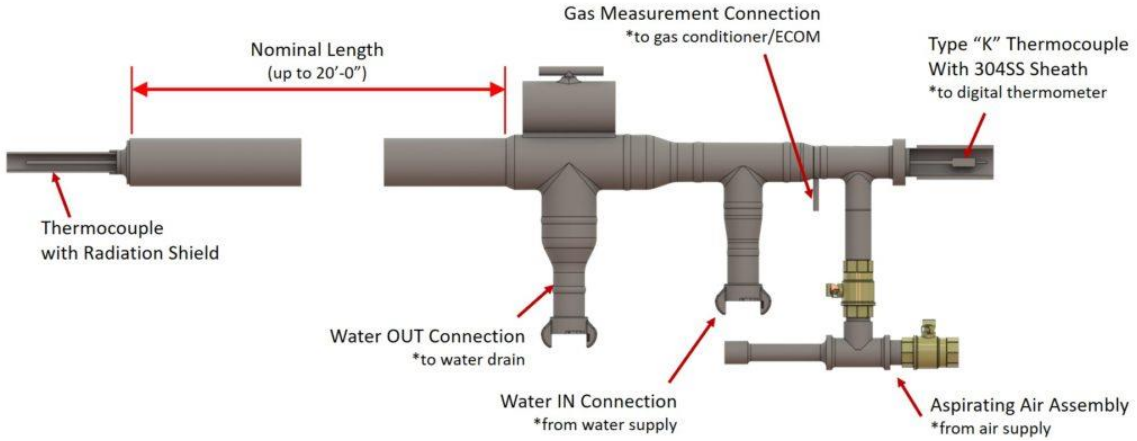


Fig. 6.7. A schematic diagram illustrating the HVT lance system used to measure flue gas temperatures and velocities in the superheater region of Plant A (reproduced from https://www.stormeng.com/ high-velocity-thermocouple-kit/).

This project's interest in the transient state of a boiler moving from one load to another offers different, time-sensitive challenges in how to sample and measure temperature. Given that commercially available probes have one sensor at their end, they can record either many times in one location or many locations over increasing time. These two factors lead to a necessary assumption that the readings the probe collect are representative of future operation, and from them the project team can develop a "signature" for the boiler: a series of conditions, or changes in conditions, where the measured outputs are expected to remain constant. The project team did not have the opportunity to record every transient, every day, for the duration of the experiment, but with overlap between external boiler conditions (e.g. load, coal), flue gas measurements (HVT lance), and the installed thermocouples in the boiler, a more complete picture can be formed; enough at least to calibrate the WVU heat transfer model in the physics based portion of the model. Under a given set of firing conditions, for the same fuel and milling operations, the flue gas temperature distribution around the superheater can be expected to remain approximately constant

over time so there should be no need to document it continuously.
The added benefit here is that it potentially allows a simpler extrapolation of this model to other boilers: develop the signature through an outage-less and inexpensive 2-day survey and use that data to power a unit-specific heat transfer model.
In Plant A, there is excellent instrumentation at the inlet and outlet of the final superheater tubing. Over 100 thermocouples record the metal temperatures of the steam flow through individual tubes, and by proxy, heat pickup by the steam. The analysis of these results clearly shows that the flue gas temperature on the edges of the boiler is greater, Fig. 6.3. However, for the sake of heat transfer model inputs, there are two unknowns: flue gas temperature and flue gas mass flow rate, and between them an infinite set of analytically acceptable outcomes. History and experience can bound those parameters to reasonable values, but still within those bounds there are too many combinations. Having physical measurements of one of those parameters helps to limit the other, within the context of the measured total heat input. For example, if during a transient, heat flux is measured to increase via the steam temperatures, it is critical to know if all locations of flue gas temperature increased a little or did a portion of it change a lot. This uncertainty is a key driver for risk when considering boiler tube longevity.
Plant A has access using a 3-inch (75 mm) port at mid-elevation just ahead of the final superheater. This location is accessible on both sides of the boiler through mirrored ports. All gas temperature measurements were performed from this access location. With the cooperation of the plant, the flue gas temperature survey using the HVT lance covered a range of behavior including different load levels (high, medium, and low) and operating conditions (before and after soot blowing).
Flue gas sampling and temperature measurement at Plant A was completed in December 2021 by Innovative Combustion Technologies. An edited version of their report is included in its entirety as an appendix, though redactions were used to mask the specific plant information. Preliminary findings indicate a significant change in flue gas temperature with operating load. A drop in power generated corresponds with a drop in flue gas temperature (Fig. 6.9). This implies a specific need for the heat transfer model to accurately identify spatial and temporal fluctuations in flue gas temperature at least within this observed range. Between the data collection on the first day and the second, the plant operated several key soot blowers which appear to have had an impact on measured flue gas temperatures, especially on the right side. Soot blowing uses auxiliary steam to dislodge collections of ash from inside the boiler and is intended to increase and balance gas flow and aid in heat transfer; however, the observations in Fig. 6.9 indicate some inconsistency.

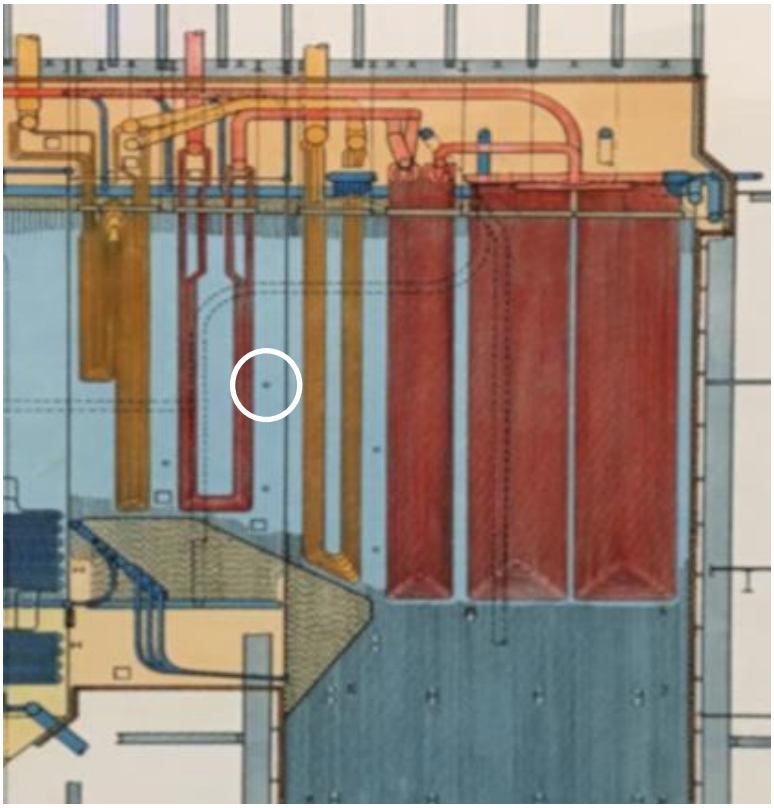


Fig. 6.8. Schematic of Plant A and the access port just upstream of the final superheater.

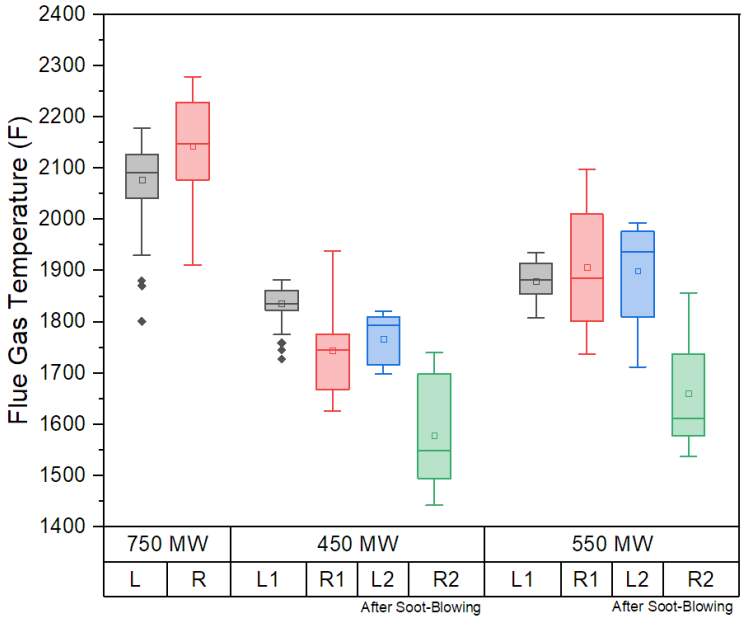


Fig. 6.9. Boxplot of the temperature data recorded by HVT at Plant A during December 2021.

As a valuable input to the "signature" analysis and in turn an input to the heat transfer model, the team has analyzed steam temperatures in the context of decreased load and operational flexibility for Plant A. These data aim to provide some insight or dependency of temperature distributions on boiler operating conditions. Here are a few main considerations:

- While the resolution of steam outlet data in Plant A is above average, it still only offers ~50 data points to cover >1100 tubes. This results in a significant possibility for missed random occurrences such as the fact that individual tube temperatures along the width of the boiler can vary significantly from their neighbors. A spike in local temperature was detected in this data set on tube 83 (Fig. 6.10), which is consistent across all load levels.
- After the outage in 2021, a significant shift in measured temperature has been recorded for approximately one quarter of the tubes. Fig. 6.10 can be compared to Fig. 6.11, both of which show low-load operation (~450MW). After the outage, the left side of the boiler shows elevated tube temperatures (+70F on inlet and outlet) which were not present before.
- Before the 2021 outage, there was no substantial change in temperature profile across the boiler due to changes in operating load. Fig. 6.11 shows the measured temperature distribution under low load, in contrast to Fig. 6.10. Similar temperature distributions were measured for other load conditions and are consistent across high, medium, and low load operation.
- In places, the characteristic shape of the inlet and outlet data changed significantly from before to after the outage. In the context of a trained analytical model, these types of unpredictable step changes may pose some challenges.

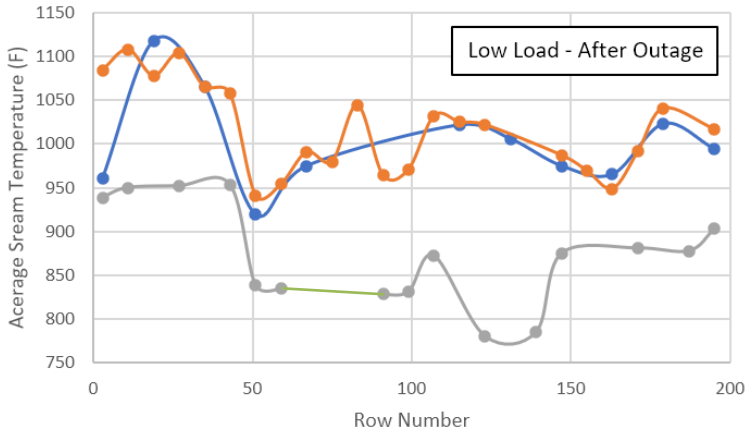


Fig. 6.10. Steam temperature measurements from the inlet header (grey) and outlet headers (outer row in blue and inner row in orange) after the 2021 outage and at low unit load.

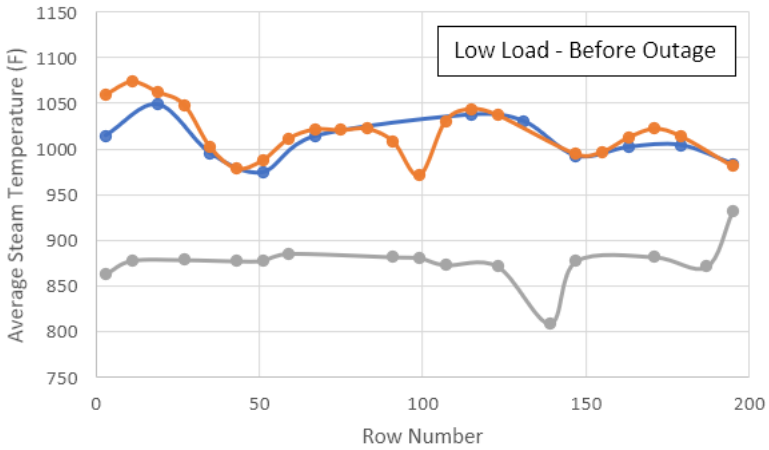


Fig. 6.11. Steam temperature measurements from the inlet header (grey) and outlet headers (outer row in blue and inner row in orange) before the 2021 outage and at low unit load.

Further analysis on these temperature profile data can be valuable in demonstrating the impact of temperature on the rate of life consumption. Feeding the temperatures into a simple creep model based on T22 material properties [1] can produce a representation of expected life consumption. This preliminary plot only considers a single static temperature over time but shows the sensitivity that life has to temperature in a different way.

At an internal pressure of 3600 psi, the hoop stress is estimated to be 5800 psi (40 MPa) for reasonable superheater tube geometries. At 5800 psi, the materials database defines a mean life at a Larson-Miller Parameter (LMP) of 39,550 (in Rankine and C=20), or 4,850,000 hours at 1022°F (550°C). However, at higher temperatures this mean life decreases rapidly.

For the sake of illustration, if the unit continues operating at the temperatures described in Fig. 6.10 for 10,000 hours, T22 tubing would be consuming life faster in regions exposed to higher temperatures, as shown in Fig. 6.12. This figure is based on calculating the time to rupture at a given temperature and stress and then expressing 10,000 hours as a fraction of that value. In the worst case in Fig. 6.12, time to failure for an average heat of T22 at 1117°F (603°C) is about 117,000 hours; said another way, 10,000 hours would have consumed 8.5% of that tube's expected life.

This demonstration highlights the sensitive nature of the life prediction and the need to capture excursions in temperature, even for relatively short durations.

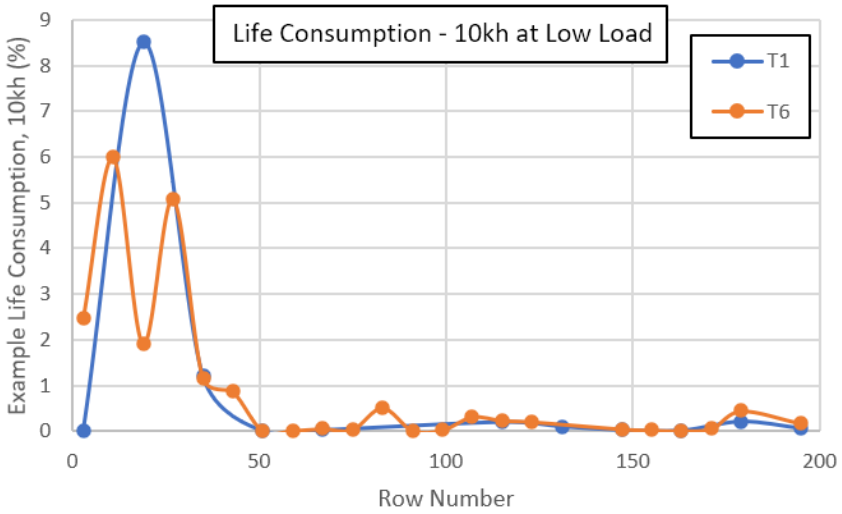


Fig. 6.12. Life consumption based on 10,000 hours at the temperatures described in Fig. 6.10

6.3 Metallography work

One pillar of this project, in conjunction with the installation of dutchmen in the plant, was the ex-service evaluation of material to correlate to the predictions from the hybrid analytical health model. Metallographic evaluation of the material after years of service is the industry standard method of understanding the degrading impact of operation and paint a picture of remaining life. In this project, the evaluation of ex-service material captures a quantitative assessment of change to the tubes which can be used to validate the predictions of the model.

6.3.1 AUSC Steam Loop from 2011

The original intent for this project was to leverage the ex-service AUSC steam loop manufactured for FG26-01NT41175 and installed in Plant A at that time as a validation exercise for the WVU heat transfer model. The various materials along the length of the loop were exposed to different temperature conditions during operation and the hope was that they would show quantifiably different metallurgical characteristics that could be correlated to an effective temperature exposure. These findings could then be used in reference to the model predictions as a way to compare and validate the outputs using real-world material. Due to several factors described below, after some investigation this effort was interrupted in favor of another avenue.

In light of the inability to accurately correlate time at temperature on the AUSC steam loop materials to observable microstructural changes, other validation databases were pursued. The project team identified another plant where significant temperature measurements were installed. Plant B, a 2-on-1 gas turbine combined cycle power plant, was the subject of additional instrumentation in the heat transfer area. These 70 additional TCs form the three-dimensional map of data against-which the WVU heat transfer model can be validated, see Section 2.2

Quantification of Sigma Phase in Super304H sections of Plant A AUSC Steam Loop

This next section describes the metallurgical evaluation of the AUSC steam loop performed under this project. For further background, refer to the AUSC Boiler Materials final report [397] .
The intent here was to take metallurgical markers of temperature exposure as a proxy for location-specific temperature predictions along the length of the AUSC steam loop. Material changes associated with the formation of secondary metallurgical phases are known to trend with temperature and exposure (service) time, but the specifics of these time, temperature, change relationships require the development of extensive databases.
The formation of phases in steels during exposure at high temperature is a diffusion-driven phenomenon and is governed by factors such as metal temperature and duration of exposure to that temperature. Hence, if the precipitation kinetics of such precipitates is established (i.e. time-temperature-precipitation relation), tracking the formation of such precipitates (volume, size, etc.), it may be possible to predict the effective metal temperature for a given service duration and can be used as a metallurgical lifing method. Such analyses have been previously performed, for example, on conventional austenitic stainless steels [398], where the effective tube metal temperature was predicted for superheater tubing by measuring the width of sigma phase precipitates present in the microstructure. To explore whether a similar approach could be applied to the Plant A steam loop, quantification of precipitate phases was performed on sections of Super304H.

Super304H Segments in Plant A Loop

Super304H is a Cu, Nb, and N added advanced austenitic stainless steels and, similar to conventional austenitic stainless steels, was found to form sigma phase at higher temperatures [399]. The plant A loop included five Super304H segments distributed at various locations along the length of the loop as shown in the Fig. 6.13 and marked with an estimated local temperature. Segments 1, 20, and 40 were selected for microstructure analysis as these segments were expected to attain three different temperature profiles as the temperature of steam increases along the length of the segment (Segment 1: lower temperature, Segment 20: intermediate temperature and Segment 40: higher temperature). Each of the three Super304H segments were machined to obtain two sections as shown in Fig. 6.14; one from a region of tube that is facing the flue gas (hotter region ‘A’ in Fig. 6.14) and the second from a region that was facing away from the flue gas (relatively colder region ‘E’ in Fig. 6.14); the temperature difference from A to E is anticipated to be on the order of 30C (50F). These sections are hereby referred to ‘Section A’ for the hotter section and ‘Section E’ for the colder sections.

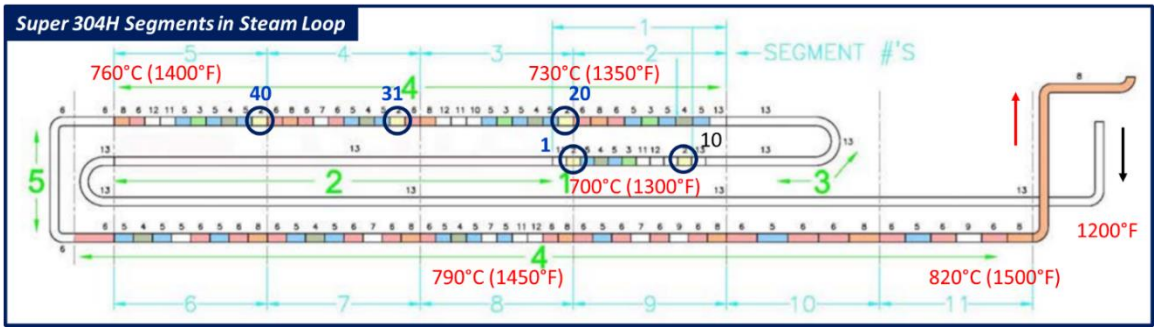


Fig. 6.13. Schematic showing the locations of Super304H segments in Plant A steam loop.

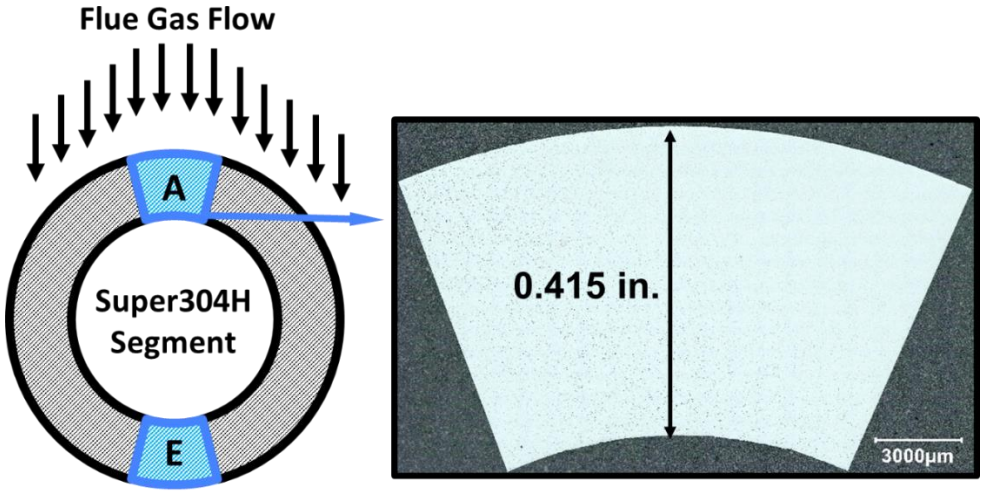


Fig. 6.14. Image showing the locations where Super304H segments from the AUSC steam loop were sectioned with respect to the flue gas flow and their appearance of the section after polishing.

Sample Preparation and Metallurgical Analysis

The six sections obtained after machining Super304H segments 1, 20, and 40, were mounted in a conductive phenolic resin and polished to a surface finish of 0.04 µm using standard metallurgical sample preparation techniques. The polished samples were then electrolytically etched using 20% KOH at 3V. Previous work has shown that this etchant highlights sigma phase in the microstructure as a straw color and an example of such an etched microstructure is shown in Fig. 6.15.

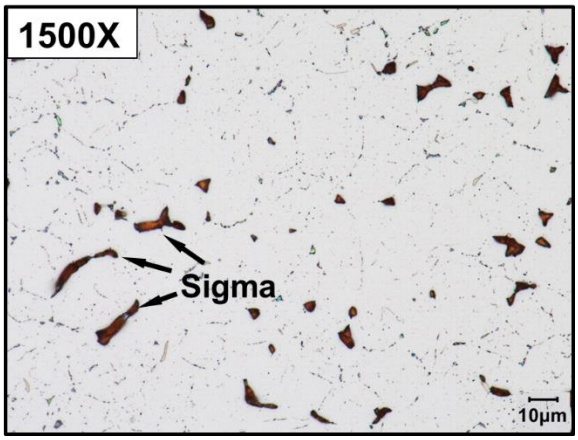


Fig. 6.15. Microstructure of Super304H after electrolytic etching with 20% KOH solution at 3V showing the presence of sigma phase.

The amount of sigma phase formed in the etched microstructure can then be quantified using image analysis techniques. An image analysis software package, ImageJ, was used to analyze the sigma phase present in the microstructure. By segmenting the etched image using the difference in the color of the matrix and precipitate features, the sigma phase regions were be highlighted as shown in the example in Fig. 6.16. A feature thresholding operation was also applied to remove features below 1000 pixels-squared to remove features such as carbides.

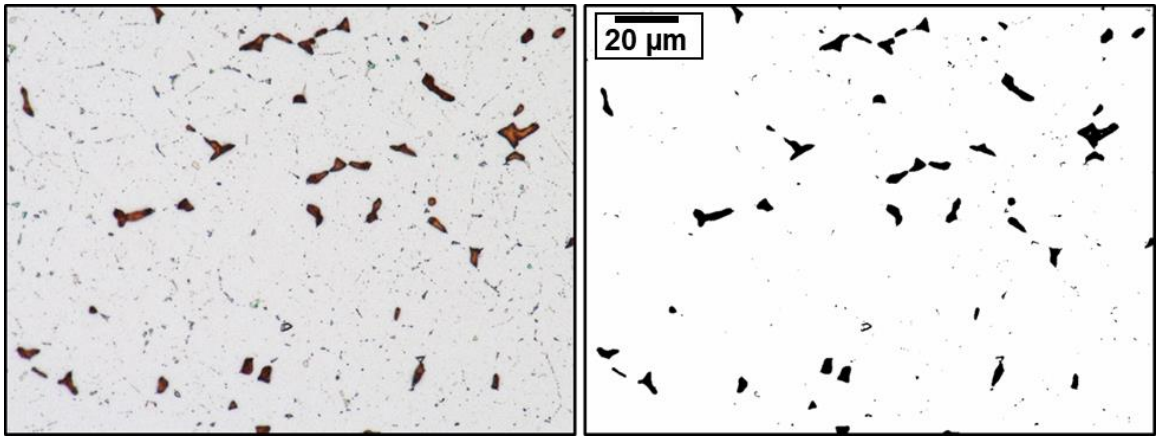


Fig. 6.16. Image of Super304H microstructure (left) and the result obtained after segmentation using image analysis sub-routine which highlights sigma phase showing about 2% sigma.

To determine the variation in the amount of sigma formed in Section A and Section E regions of the three Super304H segments, a 10 by 77 grid of microstructure images were captured from the OD to the ID regions of the two sections, as shown in the Fig. 6.16, using a Keyence VHX-7000 microscope at a magnification of 1500X. The image analysis sub-routine was used to highlight sigma from the OD to the ID of six sections of the Super304H and the results of which are plotted in Fig. 6.17.Measurements were generally between 1 and 3% and showed a consistent increase in the amount of sigma phase from the tube ID to OD. The black line in each of the six plots is obtained by joining the mean amount of sigma at each radial location and is used as a scale to study the change in sigma formed from the OD to ID regions of the six tube sections.

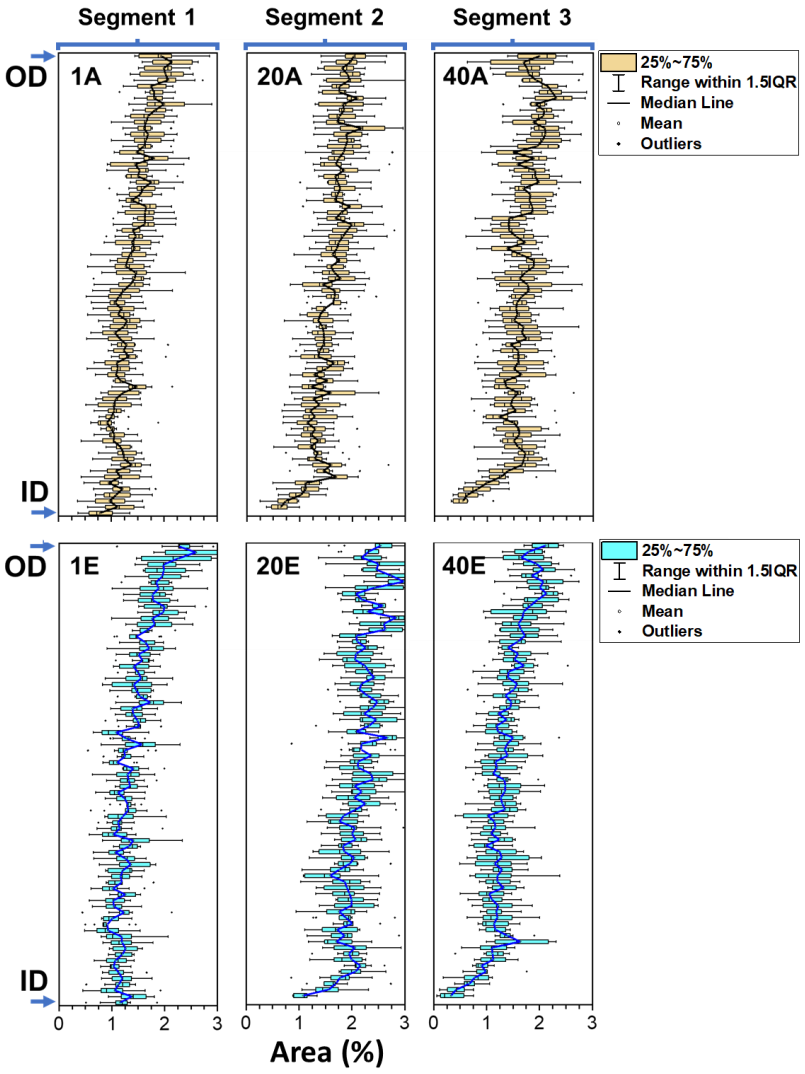


Fig. 6.17. Graphs showing the distribution of sigma phase from OD to ID regions of the six Super304H sections.

From the results shown in Fig. 6.17, the OD regions of all the sections consistently showed a higher amount of sigma than the ID regions. This suggests that the formation of sigma at the OD is fastest, which could primarily be due to higher tube metal temperatures attained at the OD during the operation of the steam loop. In Segment 1, Section A (hotter) and Section E (colder) regions appear to have a similar distribution of sigma from the OD to the ID region, suggesting very little difference in temperature between the two sides of the tube. However, contrary to expectations, Section E in Segment 20 showed a higher amount of sigma (>2%) than Section A. Currently, there is no clear explanation for this occurrence, but local changes in original composition or grain structure may have played a role. Further along the steam loop, segment 40 showed a slightly higher distribution of sigma phase (>1.5%) compared to the hotter sections of segment 1 and segment 20, indicating that the average tube temperature at the hotter regions of Segment 40 was higher than that of segment 1 and segment 20.

Clearly, based on thermodynamic phase stability calculations, Fig. 6.18, the Super 304H segments in the AUSC steam loop did not reach thermodynamic equilibrium. Thus, we have two conflicting contributions to volume fraction of sigma present in the microscopy: the rate of transformation, which is faster with increasing temperature; and the total volume percent expected at equilibrium, which reduces with temperature. When inspecting in a non-equilibrium state, this contradiction of contributions confuses conclusions about mean temperature along the length.

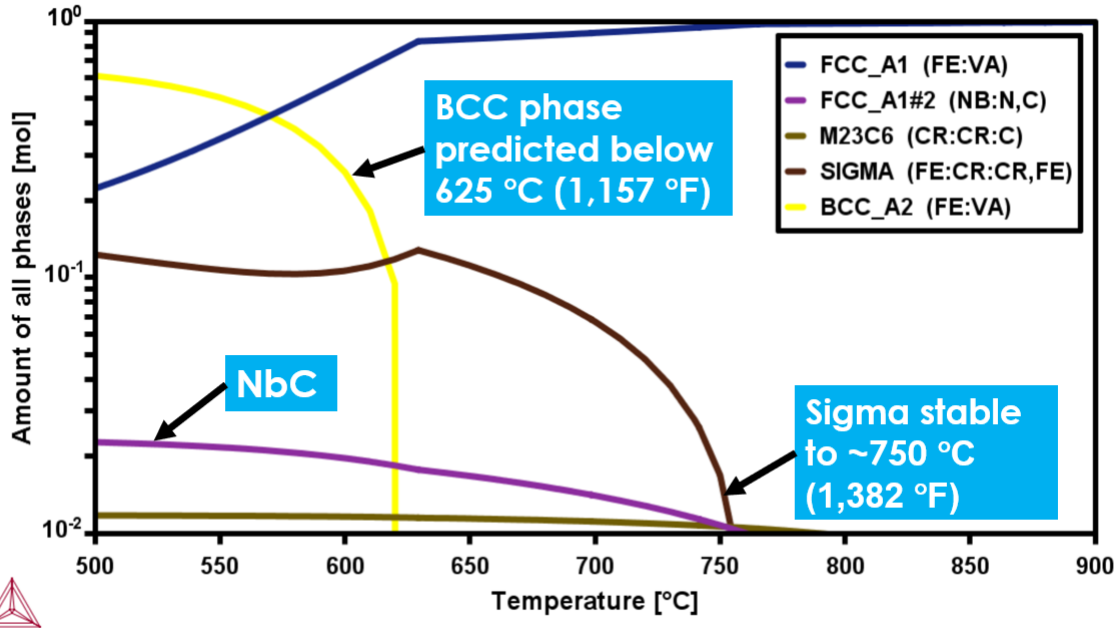


Fig. 6.18. Thermodynamic predictions from ThermoCalc of Super 304H phase stability with temperature. Note the predicted volume% of sigma with temperature.

Challenges with Correlating Sigma Amount to Temperature

Although the amount of sigma formed at the three segments was successfully determined, there were challenges in establishing the average tube metal temperature. This is primary because:

1. The sigma phase evolution kinetics for Super304H as a function of time and temperature are not well established yet. This analysis only provides a single snapshot in the history of the material, specifically after 3 years of varying thermal exposure.
2. The maximum amount of sigma (at thermodynamic equilibrium) that can form in Super 304H decreases at temperatures over 650°C. Hence, even if it is assumed that thermodynamic equilibrium has been attained at high temperature regions of the tube, the amount of sigma formed at these regions may be equal to the sigma formed at lower temperatures regions that have not attained equilibrium yet.

For these reasons, it has not been possible to estimate the tube temperatures of the different sections of Super304H. Other avenues for validating the heat transfer model beyond the AUSC steam loop needed to be considered. EPRI has leveraged opportunities within Southern Company where instrumentation in the heat transfer region of a plant may be of value. Again, the intention here is that external results can validate the predictions from the physics-based heat transfer model and any heat transfer problem that is well understood and monitored should be considered.

6.3.2 Analysis of Ex-Service 347H Tubes from Plant A

As part of the installation of T22 dutchmen, the existing lengths of tubing from the superheater were removed. These lengths were then provided to EPRI for microstructural analysis with the intent that any observations on oxide thickness could be used to inform the starting state of the heat transfer model.

Rings from tubes 51, 99, and 147 were cut from the bottom of the 8-foot length. These rings would have been approximately 2.7 m (9 feet) from the roof of the boiler. Note that there are a total of 197 tubes across the width so positions 51, 99, and 147 represent approximately ¼, ½, and ¾ of the width. The rings were prepared for metallographic examination with the intent of capturing inner diameter oxide scale thickness in four circumferential positions. Unfortunately, tube orientation relative to the boiler was not kept so the circumferential positions here are relative to surface damage indications. A summary of those measurements is provided in Fig. 6.19, Fig. 6.20, and Fig. 6.21 , with the inner oxides referring to the spinel layer and outer oxides the magnetite or magnetite plus hematite layers. Each tube indicated little average change in total oxide thickness from the upstream side (0°) of the tube to the downstream (180°). However, significant differences in oxide thickness were observed from tube 51 (~15 µm) to 99 (~36µm) and to 147 (~25 µm). Representative micrographs are provided in Fig. 6.22, Fig. 6.23 and Fig. 6.24 .

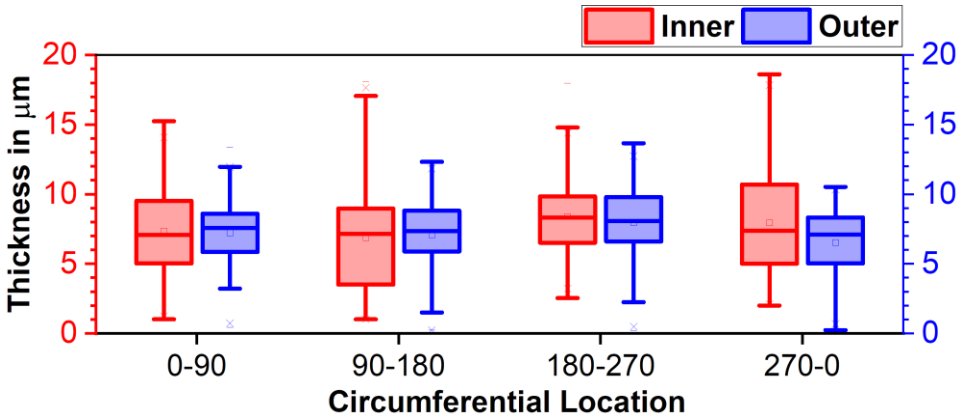


Fig. 6.19. Summary of ID oxide scale measurements taken from Tube 51. Inner refers to spinel layer and outer refers to magnetite or magnetite plus hematite.

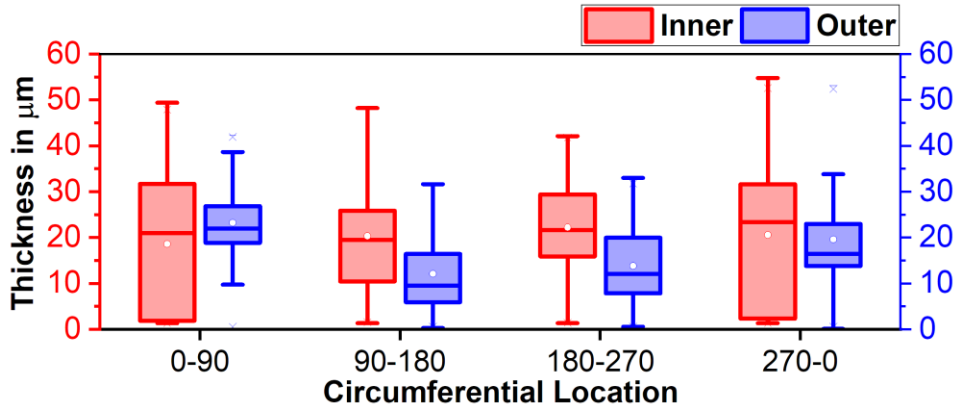


Fig. 6.20. Summary of ID oxide scale measurements taken from Tube 99. Inner refers to spinel layer and outer refers to magnetite or magnetite plus hematite.

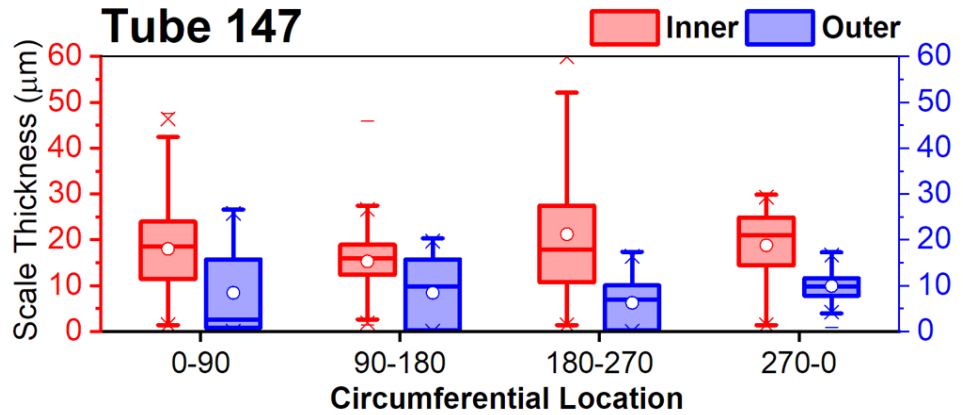


Fig. 6.21. Summary of ID oxide scale measurements taken from Tube 147. Inner refers to spinel layer and outer refers to magnetite or magnetite plus hematite.

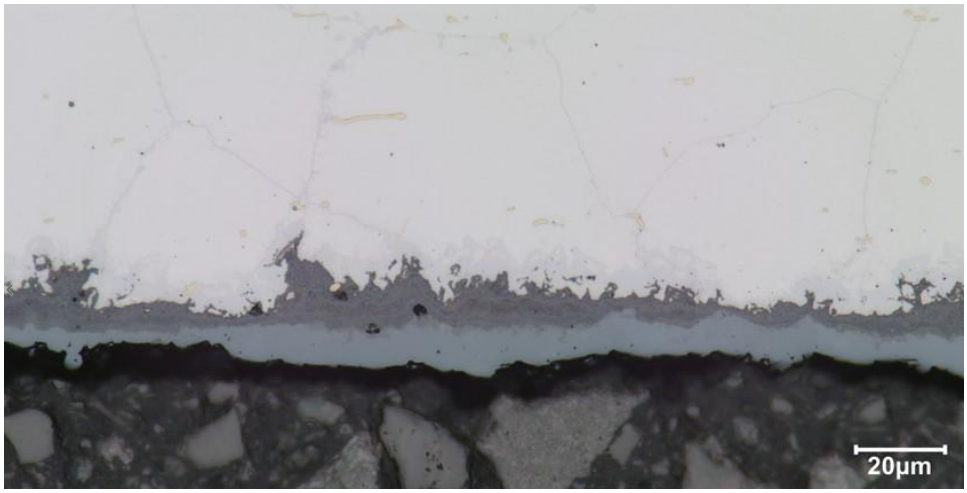


Fig. 6.22. Representative optical micrograph of ID scale on Tube 51

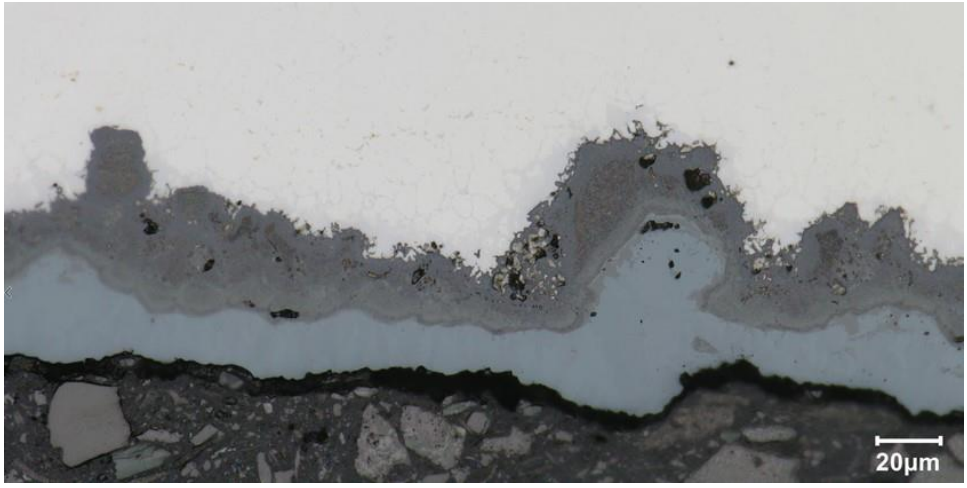


Fig. 6.23. Representative optical micrograph of ID scale on Tube 99

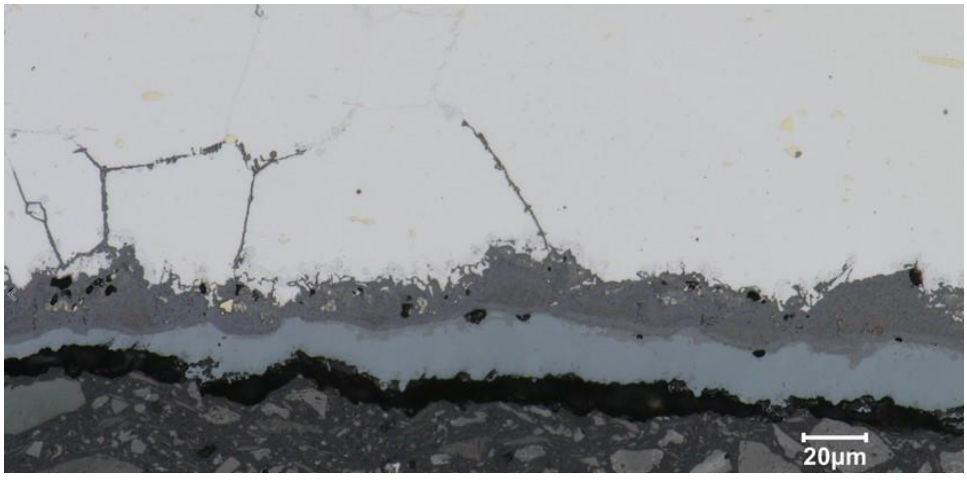


Fig. 6.24. Representative optical micrograph of ID scale on Tube 147

All of these samples show a duplex scale with the beginning of void growth with characteristic nodules associated with stainless steels. Heavy hematite is apparent and there does not appear to be evidence of exfoliation. These characteristics place the oxide in a growth state before it is thick enough to become unstable exfoliate. Of note, micrographs from tubes 51 and 147 showed some decoration of grain boundaries indicating sensitization. Sensitization is a degradation of the microstructure caused by elevated temperature excursions, likely during fabrication and not due to operation. The oxidation penetration in tube 147 is part of this damage mechanism's manifestation. If this unit were operating for much longer, this sort of damage would be a concern to longevity, but as is, it is operating at a low enough temperature not to succumb to accelerated degradation associated with sensitization in the short term.

6.3.3 Analysis of T22 Dutchmen

Analysis of the as-manufactured (spare) T22 dutchman

Six tube segments were fabricated using ASME Code Specifications for the installation of thermocouple sensors in the final superheater section of the boiler tube banks in Plant A. Replacement sections of this type are known within the boiler industry as "dutchmen". Each of these dutchmen consisted of 6-foot lengths of T22 welded to 1-foot lengths of AISI 347H stainless steel safe-ends. Use of these safe-ends which matched the superheater tubing in the plant facilitated on-site welding and installation of the dutchmen. The T22 tubing in each of the sections had dimensions of 44 mm OD (1.7 inch) and 10.5 mm wall thickness (0.41 inch).
One out of the six fabricated dutchmen was provided to EPRI for evaluation of the as-fabricated state of the tubes. Fig. 6.25 shows the dutchman tube provided to EPRI for assessment. The photograph highlights the T22 portion of the dutchman along with the two 347H safe ends.

Fig. 6.25. As-received image of one of the six fabricated dutchmen tubes showing the central T22 portion and the two 347H safe-ends. The safe ends were welded to the T22 tubing to facilitate installation in Plant A.

Following documentation, two cuts were made across the tube near the T22 to 347H safe end welds. These cuts were located such that each piece obtained consisted of a portion of T22 and 347H tubes. These pieces are shown in Fig. 6.26 a and b. Cross sectional tube rings were then cut from the T22 tubes as marked by the dotted lines in Fig. 6.26 a and b. Each ring provided a 0.25 inch thick cross-sections of the tube. These T22 ring sections were then mounted in a conductive phenolic resin and prepared for metallurgical evaluation using standard metallographic polishing techniques to a surface finish of 0.05 µm.

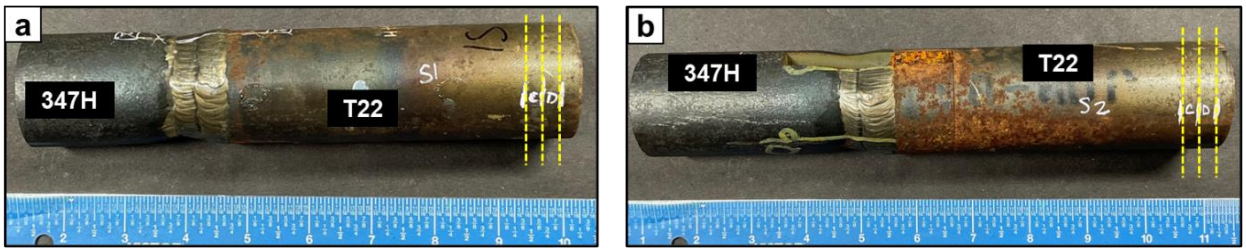


Fig. 6.26. a and b: Images of the two sections removed from the provided dutchmen showing the length of T22 tube, the length of 347H safe-end and the weld fabricated between these tubes. Cross-sections of the T22 tubes were taken from these welded sections as marked by the dashed lines.

Metallurgical analysis of T22 tube focused on establishing the base metal microstructure as well as the structure and characteristics of the surface oxide scale formed at the ID bore surface of the T22 tubes during fabrication. This baseline is considered to be the starting point of all the plant-installed tubes. Micro hardness measurements and analysis were also performed to determine if variations existed in the through-thickness plane of the T22 tubes.

The ID surfaces of the mounted metallographic samples were examined in detail using a scanning electron microscope. The focus of this examination was to investigate the surface scale structure formed in the T22 tubes during fabrication. The ID surface was imaged using the secondary electron imaging mode and a representative image of the ID surface is shown in Fig. 6.27. This figure shows the matrix of a T22 tube in the top portion of the image along with a section of the ID surface. As is evident in the image, at a high magnification the ID surface is tortuous. Additionally, there was no clear evidence for the presence of any scale on the ID surface. Hence, investigations to understand the scale structure on the ID surface suggest that there is no clear and continuous scale present in the as-fabricated state.

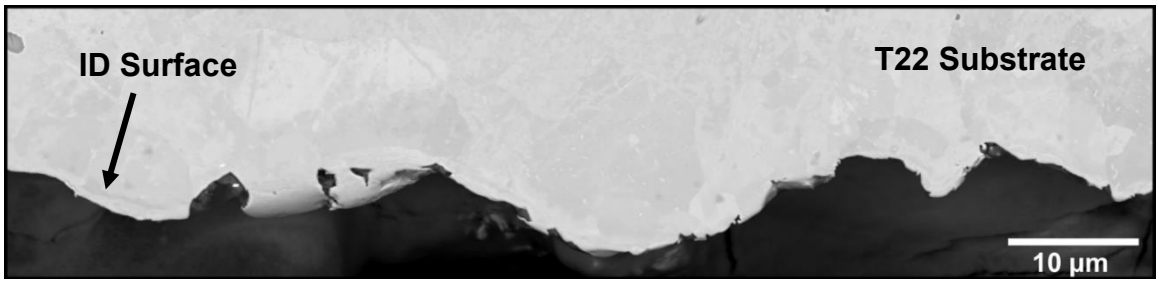


Fig. 6.27. A scanning electron microscope image of the T22 substrate near the tube ID regions showing the ID surface which had not clear evidence of any scale present in the as-fabricated state.

Ex-Service Dutchmen extracted in 2023

Five out of the six T22 dutchmen tubes introduced above, were installed in Plant A and were in service from November 2021 to October 2023. During this period the tubes experienced typical plant operating conditions including over 10,000 fired hours. After extraction, the five tubes were sent to EPRI for characterization. The primary focus of the characterization task involved the analysis of steam-side oxide scales developed at the ID surfaces of the tubes during service and developing cross-sectional microhardness distribution maps to determine the change in microhardness of the five tubes after prolonged service exposure.

As a point of reference, the as-manufactured sample was also characterized for oxides and hardness, discussed above. In summary, the as-received T22 material showed an immeasurable oxide thickness, <1 μm, and hardness measurements within acceptable parameters, the statistics of which are described in Table 6-1 below.

Fig. 6.28 shows the five tubes as received by EPRI from the plant. The tubes were designated with their respective number in the tube bank where they were installed in the plant, which were Tube 11, Tube 51, Tube 99, Tube 147, and Tube 189. The obtained tube sections consisted of a 1800 mm (6 ft) section of 44 mm (1.7 inch) OD and 10.5 mm (0.41 inch) wall thickness T22 grade tube welded to 300 mm (1 ft) sections of AISI 347H stainless steel safe ends as shown in the schematic in Fig. 6.29.

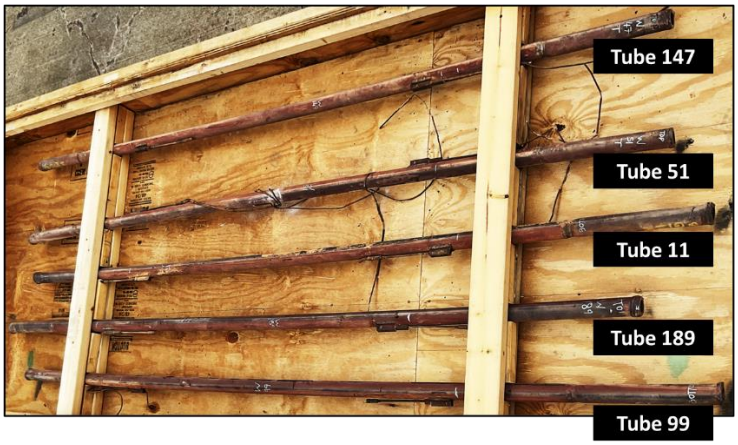


Fig. 6.28. Image of the as-received ex-service dutchmen tubes from Plant B.

For the analysis of these tubes, each tube was cut ~600 mm (~24 inches) from the hotter end as shown in the schematic Fig. 6.29 a and the obtained tube sections are shown in Fig. 6.29 b. The outer surfaces of the tubes that were facing the flue gas flow were identified based on the appearance and the deposit patterns present on the OD surfaces of the tubes. The obtained tube lengths were then further sectioned at the T22 portion of the tube section, along the highlighted dashed-lines in Fig. 6.29 b, to get two ring cross-sections from each tube. One ring section from each tube was used for microhardness testing, whereas the second cross-section was further sectioned into four quadrants for ID oxide scale analysis, as highlighted in Fig. 6.30, to prevent any possible delamination of oxides during sample preparation. The quadrants were marked as 0°, 90°, 180° and 270° to signify the quadrant's circumferential location with respect to the flue gas flow, with the 0° marking the quadrant facing upstream (of the flue gas) and 180° representing the

quadrant that is shielded from the flue gas flow. The cut sections were polished using standard metallurgical polishing techniques to obtain a final polished surface with a 0.05 μm surface finish.

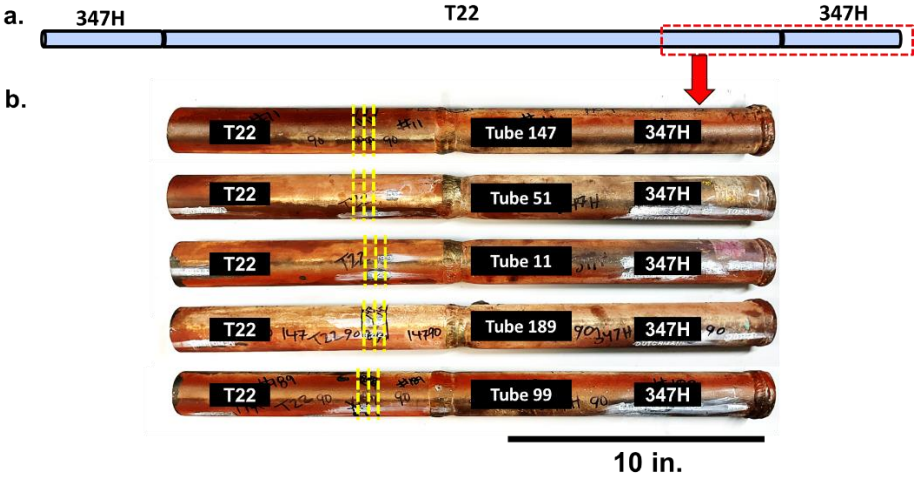


Fig. 6.29. a) Schematic of the T22 dutchmen tubes received showing 347H safe ends, and b) image showing ~24 inches long sections from the T22 dutchmen removed from one end of the as-received tubes.

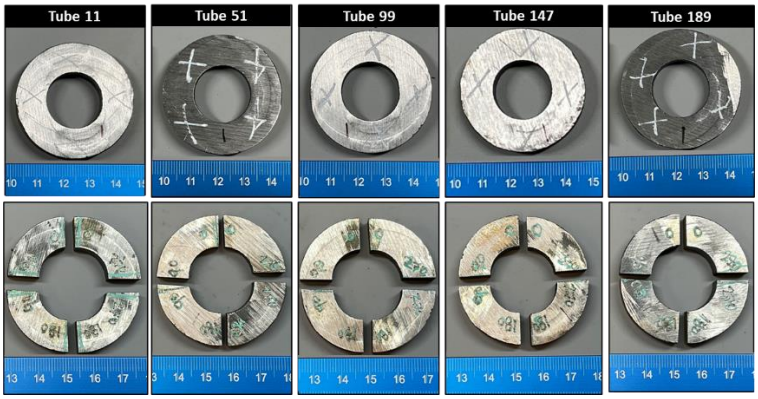


Fig. 6.30. Image of the T22 cross sections taken from the five tubes for microhardness testing (full ring sections) and ID oxide scale analysis (cut into quadrants).

Microhardness tests were performed on the polished full-ring cross sections of the five ex-service tubes using an automated Vickers microhardness machine and with a load of 500g. A grid of indents was placed on the polished surfaces using a spacing of 400 μm between adjacent indents, which resulted in ~7000 indents on the tested cross section from each tube. The obtained hardness data was plotted as a hardness distribution map using the 2D abscissa and ordinate values of each measured indent. For comparison of hardness data, a similar ring-section taken from the T22 tube in the as-manufactured condition, was also polished and tested. Fig. 6.31 shows the generated microhardness distribution maps from the five service exposed T22 tubes and from the T22 tube in the as-manufactured condition. Since the T22 tubes were manufactured from the same heat of material, it is assumed that the one section of as-manufactured T22 tube that was hardness tested represents the hardness distribution of the five tubes before they were introduced into service.

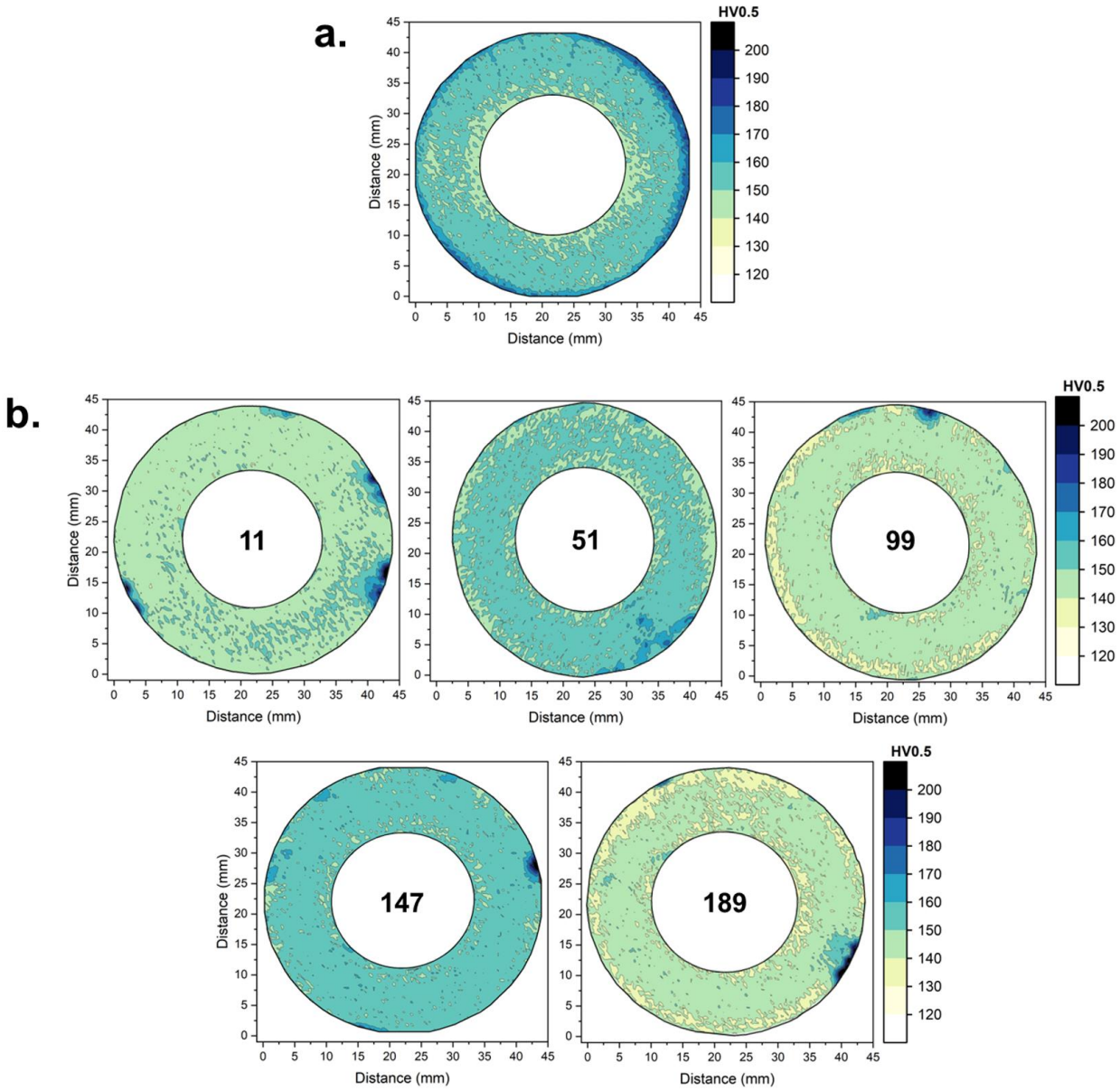


Fig. 6.31. a) Microhardness distribution map of T22 tube in as-manufactured condition and, b) image showing ~24 inches long sections from the T22 dutchmen removed from one end of the as-received tubes. 0° direction is up.

From microhardness test results, the hardness of T22 steel in the as-manufactured condition primarily ranged between 150 to 158 HV and had a median value of 153 HV, with hardness near the outer surface regions reaching 170 to 180 HV. The small surface regions of higher hardness are believed to be associated with mechanical damage and were excluded from analysis. Among the five service exposed steels, Tube 11, Tube 99 and Tube 189 showed relatively lower hardness, with median values of 147, 143 and 143 HV, respectively. In comparison, Tube 51 and Tube 147 (median values: 152 and 154 HV) did not show a marked reduction in hardness when compared with the hardness of as-manufactured tube. The reduction in hardness of Tube 11, Tube 99 and

Tube 189 suggests that these tube sections may have experienced relatively more aging possibly from higher tube metal temperature resulting from relatively higher temperature of steam flowing in these tubes during service which is comparable to the discussion in Section 2. Table 6-1 below summaries the hardness results from the six tested cross sections and a comparison of the hardness results between the six T22 tubes is shown in a box plot in Fig. 6.32 .

Table 6-1: Summary of hardness data from the microhardness testing of the six tested T22 steels.

Tube	Total number of indents	Mean (HV)	St. Dev. (HV)	Minimum (HV)	Median (HV)	Maximum (HV)
As manufactured	7066	155	7.3	136	153	192
11	6964	147	4.5	132	147	179
51	6993	152	4.1	139	152	178
99	6951	143	4.1	131	143	179
147	6984	154	3.7	140	154	179
189	6889	143	4.4	130	142	179

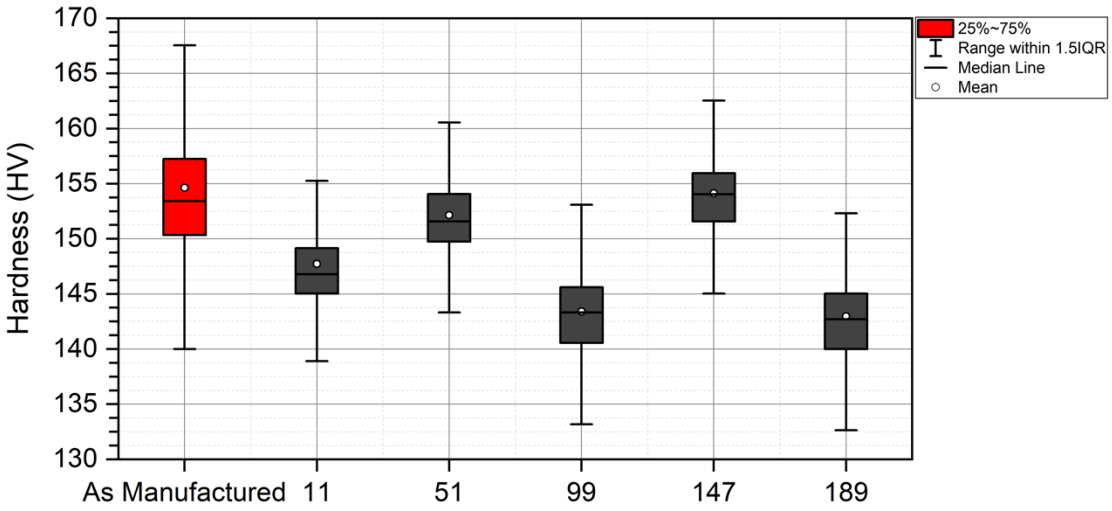


Fig. 6.32. A box plot showing the distribution of microhardness in the five ex-service T22 tubes in comparison with the hardness of T22 steel in the as-manufactured condition.

The thicknesses of oxide scales formed at ID surfaces of the five service exposed steels were determined by analyzing the tube cross sections that were cut into quadrants. Thickness measurements were carried out across the circumference of the five tubes. For this, optical images of the oxide scale preset at the ID surface of a steel were first captured along the circumference using a light optical microscope. Images were taken at 500X magnification, and each image captured a circumferential length of ~590 μm of the ID surface. Fig. 6.33 shows images of steam-side oxide scales from Tube 11 (a) and Tube 147 (b) as typical examples of the images collected from the tubes. The thicknesses of the oxide scales in these images were then measured using an image processing software called MIPAR. A representative result of a post processing operation involving the measurement of oxide scales is shown in Fig. 6.34, here Fig. 6.34 a shows the as-

capture image of the ID oxide scale and Fig. 6.34 b is the corresponding post-processed image that highlights the locations where oxide scale measurements were performed with thin vertical lines. Approximately 200 measurements of scale thickness values were obtained from each captured image, which also helps provided information on the scale thickness variation.

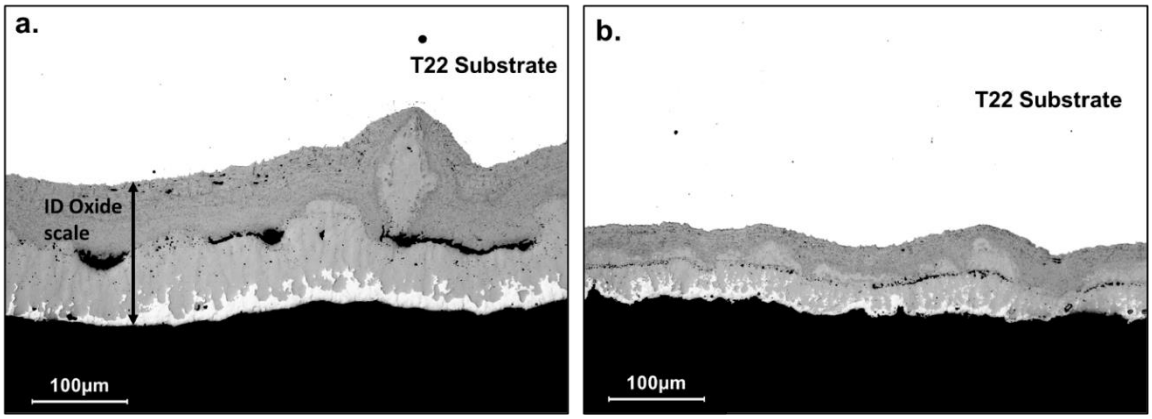


Fig. 6.33. Representative optical images of steam-side oxide scales captured at 500X magnification from a) Tube 11 and b) Tube 147.

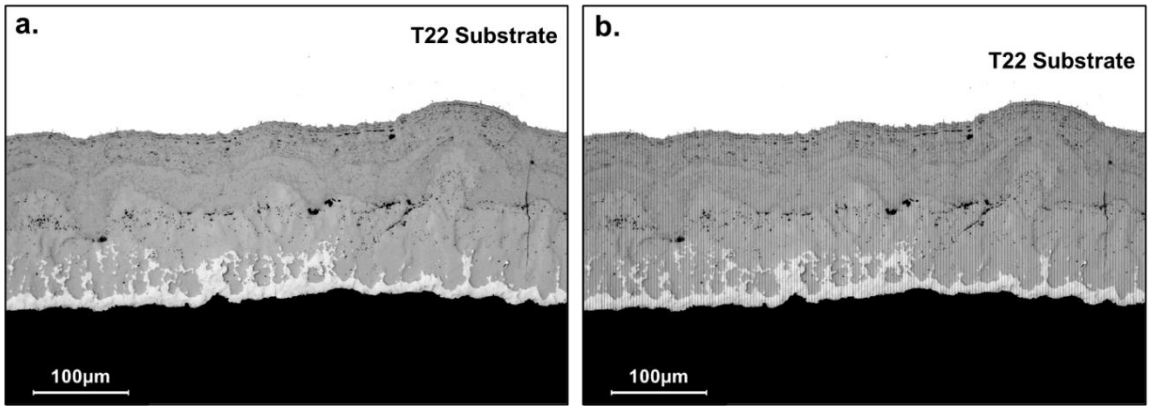


Fig. 6.34. Images showing a) an as-captured image of the steam-side oxide scale from Tube 189 and, b) after image post-processing operation where scale thickness were measured at multiple locations along the width of the image.

Steam-side oxide scale thicknesses were measured for the scales present at four quadrants of each of the five tubes. Example images of the ID oxide scales from four quadrants for each tube along are shown below along with the results of scale thickness measurements plotted as box and distribution plots.

Images of steam-side oxide scales from the four quadrants of Tube 11 are shown in Fig. 6.35a and the measurements of oxide scale thicknesses from these images are shown in the graph in Fig. 6.35 b. Most scale surfaces studied showed signs of partial scale delamination from exfoliation events, which is evident from the loss of the outermost scale regions (magnetite and hematite regions), as marked in the images below. Furthermore, regions with nodular scale growth, where the scale

growth rate is higher, resulting in localized zones with greater scale growth rate and evident scale ingression into the substrate, were also observed. This loss in oxide scale due to exfoliation and the presence of oxide scale nodules, would cause large variations in the distribution of oxide scale thickness. This is observed in the distribution of measured oxide scale thicknesses in the graph in Fig. 6.35 b, where large changes in the scale thickness are evident at 0°, 90° and 270° locations due to scale exfoliation and formations of oxide nodules. The thickness measurement obtained from 180° quadrant had a marked narrow range as this site showed almost no signs of exfoliation (presence of hematite layer) despite showing cracks in the oxide scale. This region was also found to have lower extent of nodular formations.

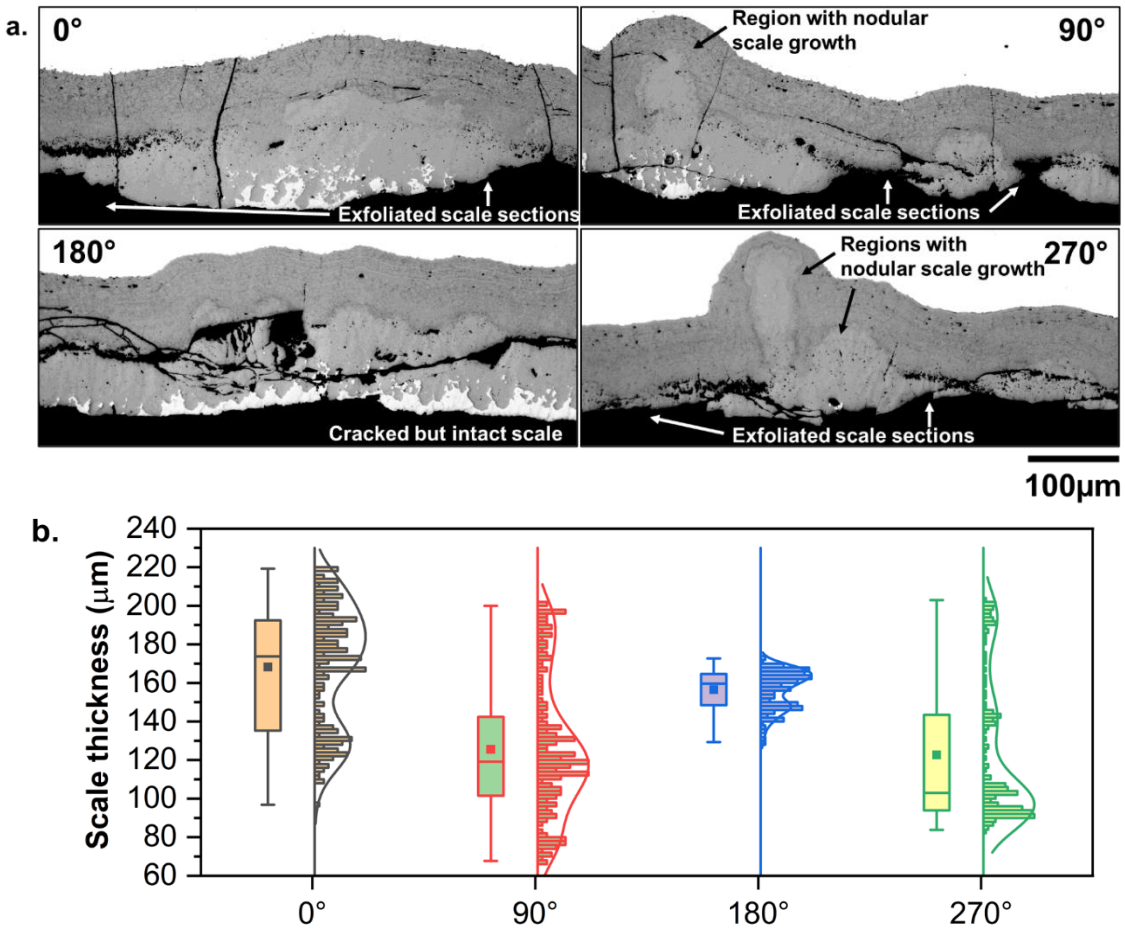


Fig. 6.35. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 11 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 11.

Images of steam-side oxide scales from the four quadrants of Tube 51 are shown in Fig. 6.36 a and the measurements of oxide scale thicknesses from these images are shown in the graph in Fig. 6.36 b. Unlike Tube 11, the oxide scales observed in Tube 51 showed less signs of exfoliation, although some observed scale regions did show signs of scale cracking. This observation is also reflected in the narrower distribution in the scale thickness measurements from the 0°, 90° and 180° locations, with most thickness measurements lying between 80 to 90 µm. However, the

measurements from the scale present at the 270° quadrant showed a wider distribution in scale thickness due to the presence of local nodular formations that result in higher scale thickness measurement values.

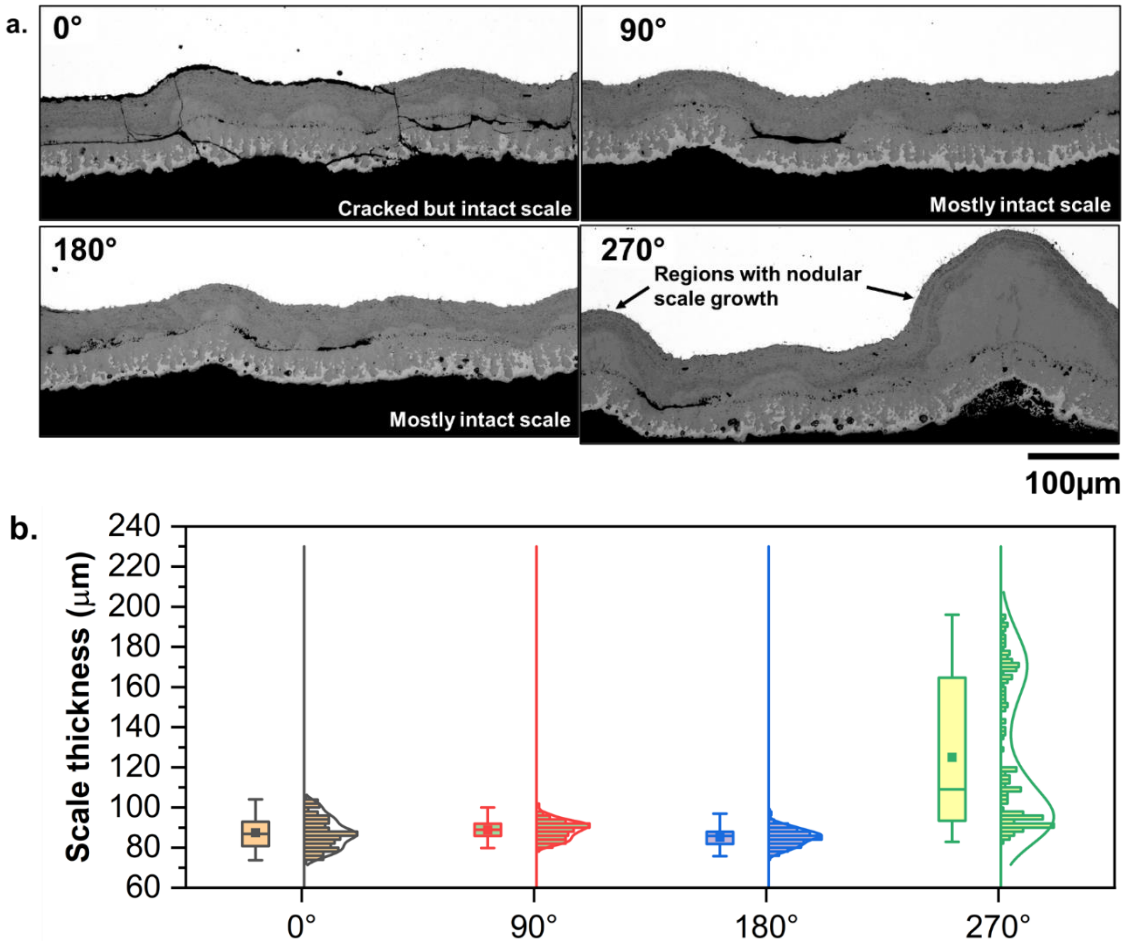


Fig. 6.36. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 51 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 51.

Images of steam-side oxide scales from the four quadrants of Tube 99 are shown in Fig. 6.37 a and the measurements of oxide scale thicknesses from these images are shown in the graph in Fig. 6.37 b. Similar to Tube 51, the oxide scale thickness in Tube 99 was relatively more uniform across the circumference and did not show many sites with signs of scale exfoliation. Most measured scale thickness values lay between 110 to 120 µm. However, local regions with nodular scale formations were also present in the oxide scales of this steel and prominently represented in the wide range in the thicknesses measured at the 90° of this tube.

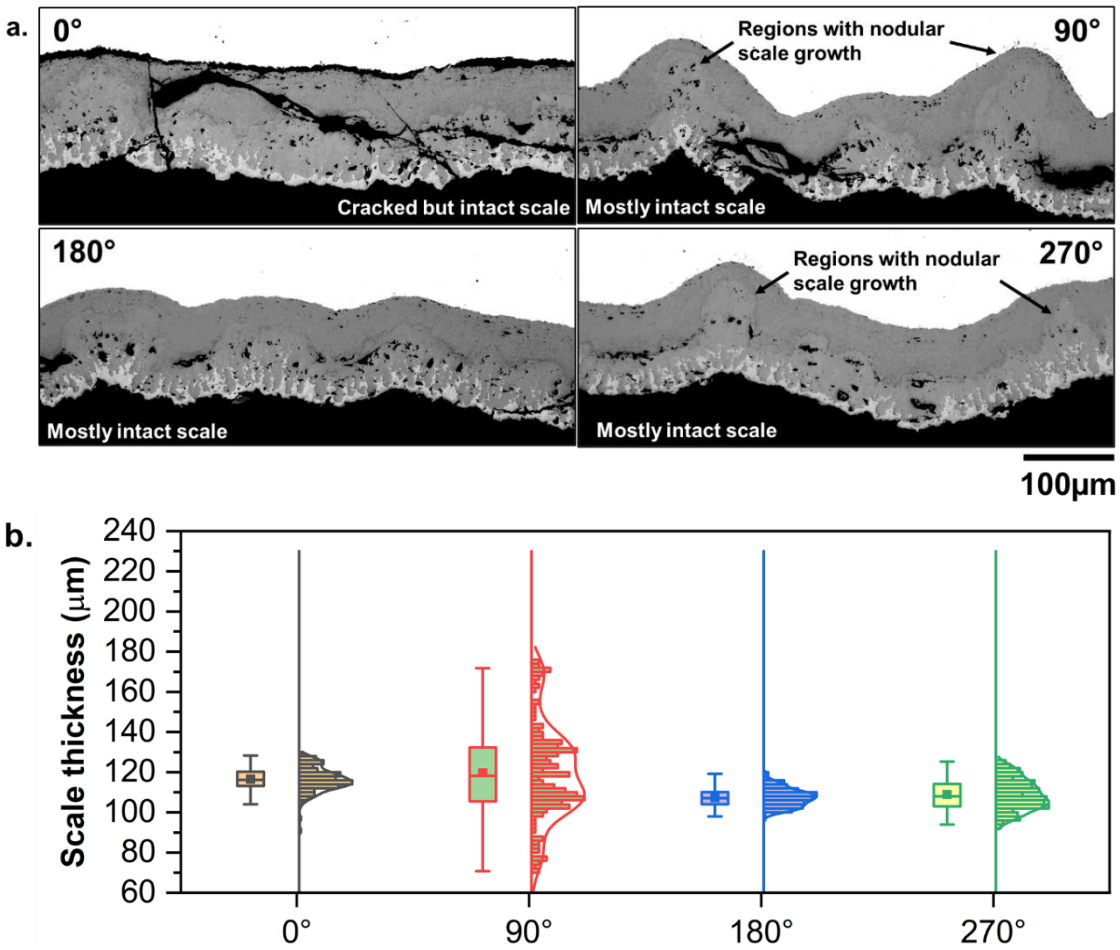


Fig. 6.37. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 99 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 99.

Images of steam-side oxide scales from the four quadrants of Tube 147 are shown in Fig. 6.38 a and the measurements of oxide scale thicknesses from these images are shown in the graph in Fig. 6.38 b. Similar the observations from oxide scale formed in Tube 51 and Tube 99, the scales in Tube 147 had a narrow range in the distribution of scale thickness suggesting a predominately typical oxide scale growth observed in Grade 22 steels and more uniformity in thickness across the circumference of the tubes. Most scale thickness values measured ranged between 70 to 90 µm. However, as was observed in the other tube sections, Tube 147 also had regions with nodular scale growth. These scale formations are highlighted in images taken from the 0°and 270° locations.

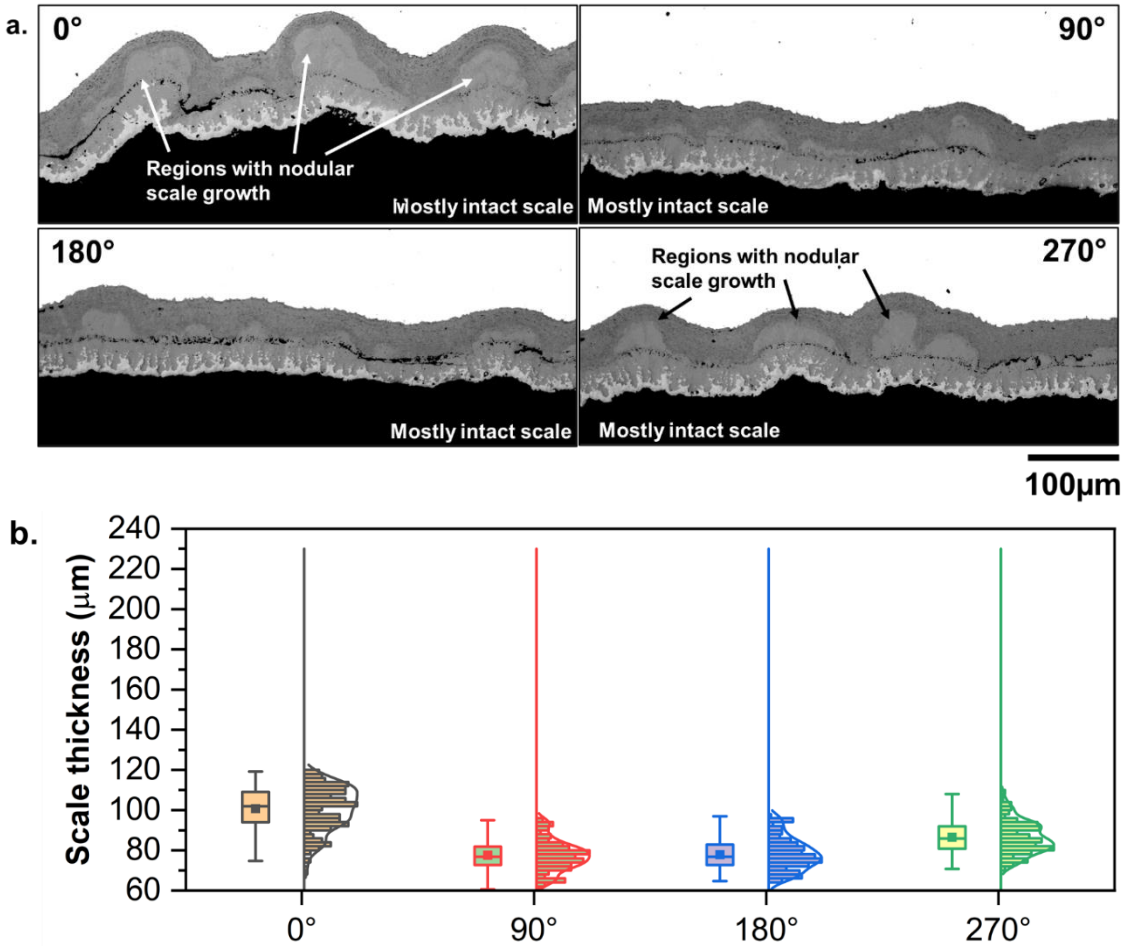


Fig. 6.38. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 147 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 147.

Images of steam-side oxide scales from the four quadrants of Tube 189 are shown in Fig. 6.39 a and the measurements of oxide scale thicknesses from these images are shown in the graph in Fig. 6.39 b. The oxide scales present in this tube also did not show significant signs of oxide scale exfoliation. Furthermore, the scales were also uniform in thickness across the circumference, with only the scale at 90° showing more indications of variation in scale thickness, likely from the local oxide scale nodules that were observed in the oxide scales in this quadrant. The thickness values measured of the scales with a more typical, uniform growth morphology was in the range of 180 to 200 µm. Fig. 6.40 is a consolidated graph that shows the oxide scale thicknesses measured at the four quadrants of the five tubes easy comparison.

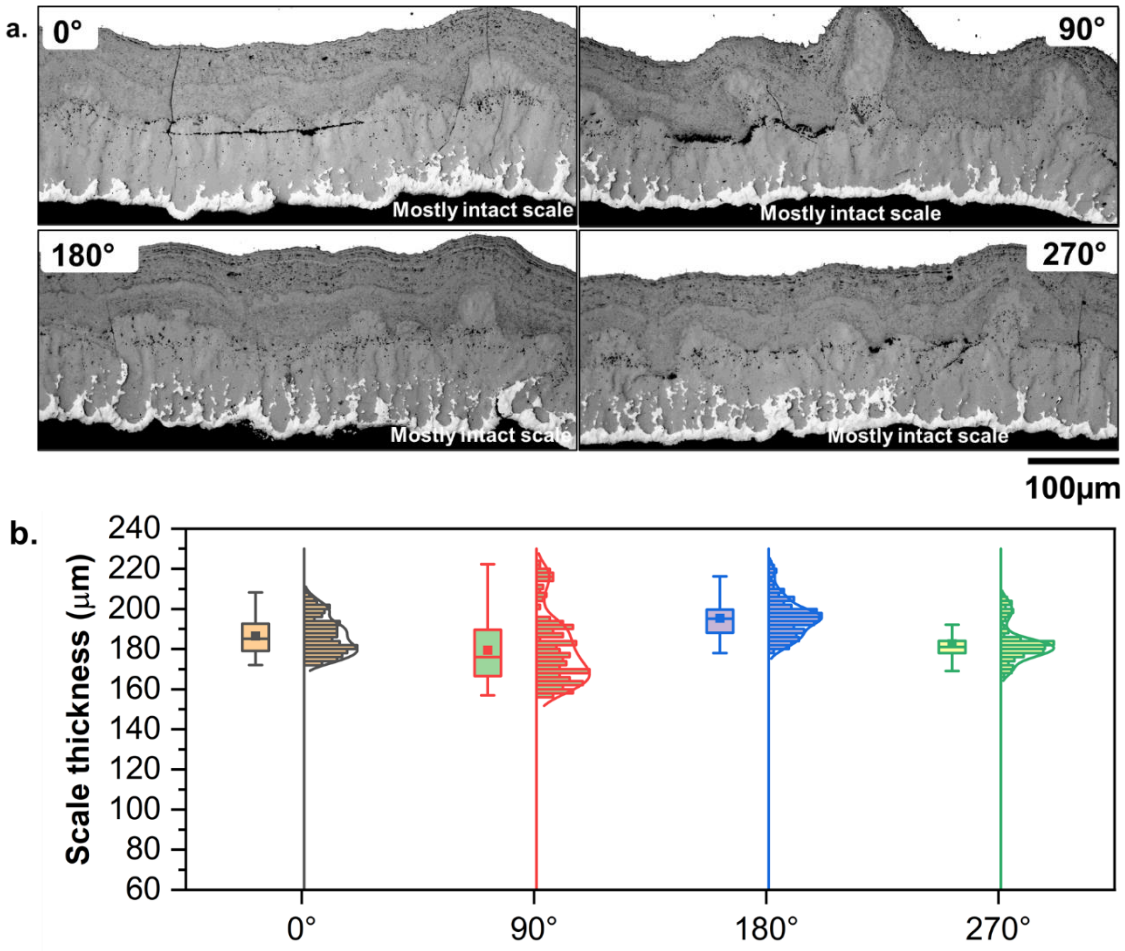


Fig. 6.39. Images showing a) an as-captured images of the steam-side oxide scales from four quadrants of Tube 189 and, b) measured scale thicknesses and its distribution at the four quadrants of Tube 189.

The comparison graph of steam-side oxide scale distribution in Fig. 6.40, shows large variations in the measured thicknesses of the oxide scale across the circumference of Tube 11 and all other tubes showing a more uniform distribution in the scale thickness except at locations were large nodular scale formations were evident. However, considering that the scale thickness across the circumference of tubes that showed little signs of exfoliation (Tube 51, Tube 99, Tube 147 and Tube 189) were uniform, and that only regions having more signs of nodules caused a larger scatter in scale thickness values in these tubes, a further data filtering step was applied to discard measurements at sites with clear signs of exfoliation and regions with high thickness readings due to oxide scale nodules. For this, only scale thickness measurements from sites with a typical oxide scale growth morphology were considered. The resulting oxide scale distribution in the five tubes is summarized in the graph in Fig. 6.41. Since the scale thicknesses are found to be uniform across the circumference, the presented data no longer includes any positional information along the circumference of the tubes. The data presented in the graph in Fig. 6.41 has been summarized in Table 6-2.

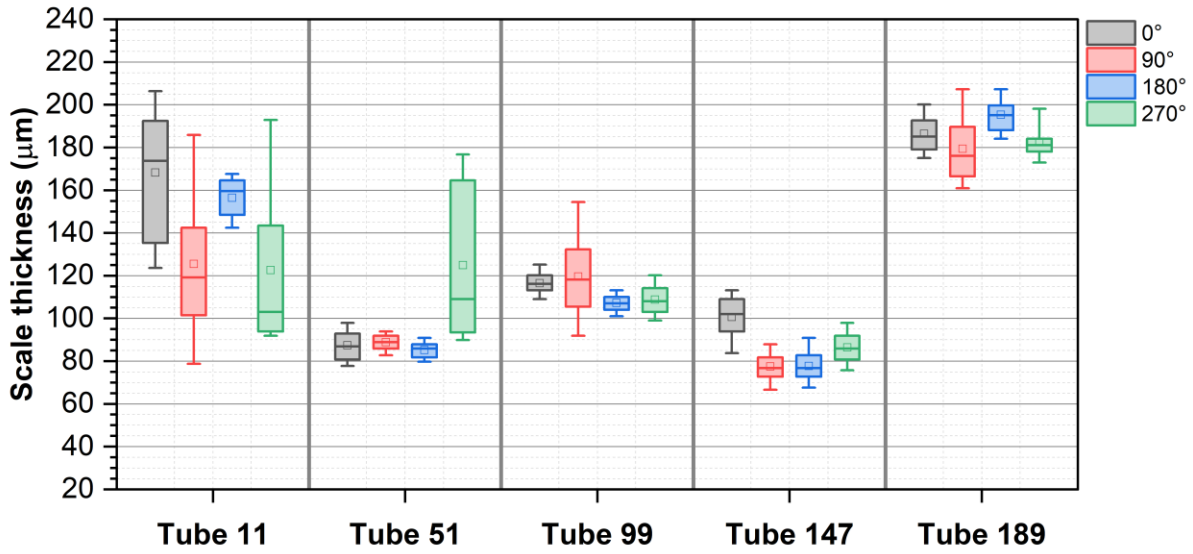


Fig. 6.40. Box plot showing the distribution of oxide scale thickness across the circumference of the five tubes studied in this work.

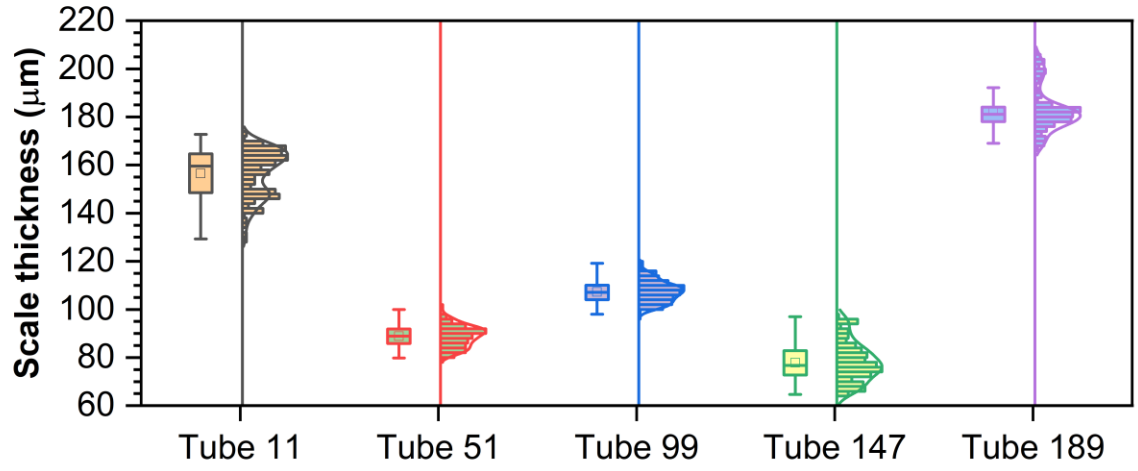


Fig. 6.41. Box plot with the distribution of steam-side oxide scale thickness measurements from the five T22 tubes.

Table 6-2: Summary of steam-side oxide scale thickness measurements from the five T22 tubes.

	N total	Mean (µm)	St. Dev. (µm)	Minimum (µm)	Median (µm)	Maximum (µm)
Tube 11	191	156.5	10.0	129.3	159.6	172.7
Tube 51	189	88.9	4.4	79.8	88.9	100.0
Tube 99	191	107.3	4.4	98.0	107.1	119.2
Tube 147	193	77.8	9.3	31.3	76.8	97.0
Tube 189	192	182.7	9.0	167.0	181.1	206.2

From the filtered steam-side oxide scale measurement data in Fig. 6.40 and Table 6-2, the scale thickness in Tubes 51 and 147 were found to be markedly lower (average of 88.9 μm and 77.8 μm, respectively) than the scale thicknesses measured from the remaining three tubes. In some cases, these measured values from Tube 51 and 147 were nearly half of the values measured (Tube 11 and Tube 189). These two tubes also had higher average hardness (152 HV and 154 HV, respectively) than the three remaining tubes. These findings suggest that Tube 51 and Tube 147 may have been operating at a relatively lower tube temperatures than Tube 11, Tube 99, and Tube 189. Estimates of tube metal temperatures of these tubes made using both the mean and the maximum values of measured oxide scale thickness [396], assuming a total service duration of 10,000 hours, are shown in Table 6-3.

Table 6-3: Estimates of tube metal temperature calculated from the mean and the maximum values of measured steam-side oxide scale thicknesses of the five tubes.

	Mean Oxide Thickness (μm)	**Est. Temp. from Mean Oxide Thickness**	**Max. Oxide Thickness (μm)**	**Est. Temp. from Max. Oxide Thickness**
Tube 11	156.5	554°C, (1029°F)	172.7	559°C, (1038°F)
Tube 51	88.9	526°C, (979°F)	100.0	532°C, (990°F)
Tube 99	107.3	535°C, (995°F)	119.2	540°C, (1004°F)
Tube 147	77.8	520°C, (968°F)	97.0	530°C, (986°F)
Tube 189	182.7	562°C, (1044°F)	206.2	568°C, (1054°F)

The estimates of metal temperatures of these five tubes suggest that Tube 11 and Tube 189 were likely operating 35 to 40C higher than the tubes with estimated lower temperatures.

Creep results (as-manufactured and ex-service)

In addition to metallographic analysis of the as-manufactured and ex-service T22 tubes reported on previously, specimens for creep testing were also manufactured and tested from these tubes. The creep testing was used as an additional way to evaluate in-service temperature exposure; post-exposure creep testing is an established approach for determining the level of creep damage during service. Operation under elevated temperature and stress will consume some fraction of the total creep life of that material. Given all the installed dutchmen have the same geometry and steam pressure, the change in post-service creep test duration can be correlated to local operating temperature. This will offer another indication of the local in-service temperature which will be used to evaluate the estimates made using the WVU models.

It bears mentioning that this creep test comparison method is not expected to be quantitative. There are several significant variables around the tube material properties and where this heat of T22 sits relative to existing databases which obfuscate a true quantitative measure of exposure temperature. However, it is believed that a relative ranking is of value.

All of the creep testing (as-received as well as ex-service) was performed at 600°C and 90 MPa at one commercial lab. Four tests were performed on the as-received material as well as duplicate tests from each ex-service tube. The results of the creep testing are presented in Table 6-4.

For a given tube the results for the repeat tests were within the expected scatter for samples made from T22 steel. However, the durations measured do not follow the expected trend. The longest lives were recorded for the ex-service tube 147. In addition, ex-service tube 51 with measured lives

of about 630 hours also showed greater creep life than the new tube section, average life 551 hours. Tubes 11, 99, and 189 showed lives less than recorded for the tests on the as-received tube.

Table 6-4: Duration of creep testing performed on ex-service T22. All tests performed at 600°C and 90 MPa.

	Rupture Times (h)				Average
As-Received	572	578	545	508	551
Tube 11	298	464			381
Tube 51	633	638			636
Tube 99	431	388			410
Tube 147	1050	977			1014
Tube 189	297	384			341

Hardness variation is one potential source of variation within the tube which may play a role explaining the discrepancies between the expectation that the as-received (unexposed) material has the longest creep rupture time. As reported on previously, each of these tube segments were tested for hardness (>100 Vickers indents each). Fig. 6.42 plots the average rupture time against average hardness, it also includes the service exposure temperature for each data point.

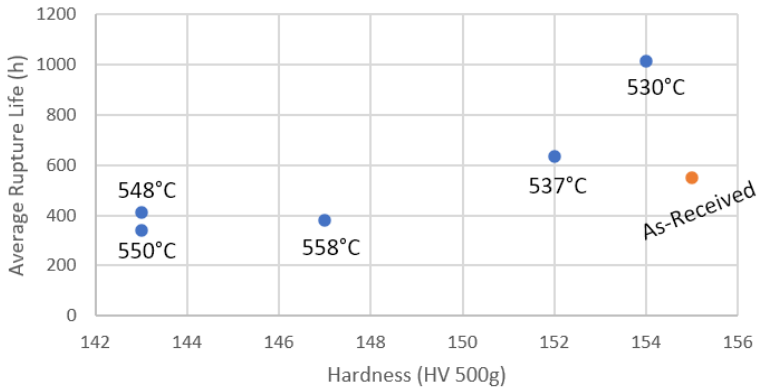


Fig. 6.42. A plot of tube test results showing average hardness and average rupture time**.**

How to interpret this data is not clear. High hardness increases creep rupture time, particularly for short (<1000h) tests. High exposure temperatures can have a softening effect, reducing a material's hardness. The figure shows a correlation between hardness and rupture test life; however, a difference of 10 Vickers is not a large variation. The figure also shows that the average exposure temperature (penthouse thermocouple data) does not completely correlate with either other metric. There is potentially some conflation of effects present between hardness variations, creep strengths, and in-service exposure clouding the interpretation of this data.

6.3.4 Discussion on Temperature Exposure and Correlation of Various Indicators, etc.

In the previous sections, particularly the metallographic analysis, the ex-service tubing was analyzed using several different methods. Each of these was selected to shed some light on the in-service conditions local to each tube. This section will compare and contrast the results from each of those methods. The data analysis methods used here compare:

- local average exposure temperature in the penthouse (~10,000h of data)
- local average exposure temperature at each dutchman TC (much less data)
- metallographic measurements of hardness (thousands of indents)
- metallographic measurements of oxide thickness (hundreds of individual measurements)
- equivalent average temperature which would have resulted in the same oxide thickness [1]
- and creep test duration as an estimate of remaining life after service

Table 6-5 The following table summarizes several indicators of exposure temperature. Note, each of these are absolute measurements at the true location, not corrected for any expectation of heat flux*. Further, given some observed discrepancies, it bears mentioning that the penthouse TC locations were explicitly rechecked to be correctly labeled – and correctly attached to the same tube. It is important to mention that the heat flux condition of the tubes in the penthouse differs from that of the dutchmen as the dutchmen were situated in the boiler. The penthouse TCs measure the tube outlet steam temperature. The dutchmen TCs, while they lasted, measured the tube exterior surface temperature, subject to heat flux. The oxide thickness measurements approximate the metal temperature at the ID surface of the tube. The dutchmen creep testing, were these dutchmen exposed to a significant fraction of their life, could be used to indicate the mid-wall temperature by means of approximating consumed life during operation relative to remaining life after extraction; however, given that two of the ex-service tubes lasted longer than the as-manufactured tubes, it becomes impossible estimate life consumed in this way.

*It is estimated that under the conditions in the heat flux region of the boiler, the heat flux would be expected to cause a temperature gradient from OD to ID 20C (before any oxide develops). As oxide grows, it resists heat transfer, and so the gradient is expected to increase. Thus we would expect that the dutchmen TC (OD) readings should be significantly higher than steam temperature readings in the penthouse. Steam temperature readings would be expected to be very close to the ID oxide growth-inferred temperature.

The penthouse TCs measure the tube outlet steam temperature as there is nearly no heat flux in the penthouse. The dutchmen TCs, while they lasted, measured the tube exterior surface temperature in a region of heat flux. The oxide thickness measurements approximate the metal temperature at the ID surface of the tube. The creep testing of the ex-service dutchmen material, were these dutchmen exposed to a significant fraction of their life, could potentially be used to indicate the mid-wall temperature of the tube.

Table 6-5: Summary of data and sample analysis for each of the five locations

	Right	CR	Center	CL	Left
Dutchmen Assembly Number	**11**	**51**	**99**	**147**	**189**
Dutchmen TC Hours (h)* over 480°C	686	687	1405	687	891
Dutchmen Average (OD Surface) Temperature (°C)*	591	594	558	581	551
Dutchmen Oxide Scale Thickness (μm)	157	89	107	78	183
Calculated Exposure Temperature from Oxide Thickness at 10,000 hours (°C)	554	526	535	520	562
Dutchmen Average Hardness (HV)**	147	152	143	154	143
Dutchmen Average Creep Test Duration (h)	381	636	410	1014	341
Penthouse Assembly Number	**11**	**51**	**99**	**147**	**195**
Penthouse TC Total Hours (h) over 480°C	10003	9981	9993	9980	9995
Penthouse Average (Steam) Temperature across entire exposure (°C)	558	537	548	530	550
Penthouse Average Temperature over how long the Dutchmen TCs lasted (°C)	555	534	548	526	551

*Two TCs were attached to each dutchman. This table reports the longer-lasting total hours and the time-weighted average of the two average temperatures.
**Hardness is included in this table as an indicator of in-service tempering and also because at short creep test durations hardness can have a significant influence on rupture time.

Table 6-5 shows several metrics comparing the impact of exposure to the tubes. To summarize this information, the next table presents the same metrics normalized to a simple ranking of hottest/thickest oxide (1) to coldest/thinnest oxide (5) at assembly locations relative to the boiler gas path.
There is strong agreement between the measured oxide scale thickness and the penthouse temperature as well as the ex-service creep rupture time. The dutchmen (surface) temperatures align poorly with the other indicators – with a reminder that the dutchmen temperature readings only represent a small fraction of the entire duration of the exposure before the TCs failed. Whereas, the penthouse TCs continued reading for the entire duration. Further, the dutchmen readings did not generally correlate well to the penthouse temperatures over the timeframe where they could be compared.
Creep rupture duration ranking for each location also correlated well with the oxide thickness ranking. Given the short duration of the dutchmen TCs recording accurate measurements, the most trustworthy indicators are assumed to be the physical measurements of oxide thickness and the penthouse TCs. These two indicators do show good agreement in the ranking of hottest tubes.

Table 6-6: Ranking of select data from Table 6-5 for each of the five dutchmen locations

	Right	CR	Center	CL	Left
Dutchmen Average (OD Surface) Temperature	2	1	4	3	5
Dutchmen Oxide Scale Thickness	2	4	3	5	1
Dutchmen Average Hardness (reversed)	3	4	2	5	1
Penthouse Average (Steam) Temperature (10kh)	1	4	3	5	2
Dutchmen Average Creep Test Duration	2	4	3	5	1

6.4 Gas-Touched Length Analysis (GTLA)

The term "Gas-Touched Length" (GTL) herein refers to segments of a steam circuit in a steam generator that are exposed to an external heat source, such as flue gas, for the purpose of heat transfer to the steam flowing within the circuit. In a typical steam generator, the steam circuits which have segments involved in heat transfer from an external heat source, hence constituting "gas-touched length" segments, are superheaters and reheaters. Analysis performed to predict the spatial distribution (typically radial and along the length of each tube, and for multiple tubes across the width of the steam generator) of steam and tube metal temperature over such a "gas-touched length" segment of a circuit is herein called "Gas-Touched Length Analysis" (GTLA).
A similar analysis is typically performed during the design stages of the steam generator where the desired process parameters, such as outlet steam temperature, pressure, mass flow rate, and heat flux, are known. The analysis involves calculating the overall heat transfer rate (or heat flux) from the heat source, through the tube wall, and to the steam. This requires various inputs based on the:

- external heat source, such as the temperature and heat transfer coefficient for flue gas flowing over the gas-touched length segments
- tube geometry, such as the diameter and wall thickness, and tube material parameters, such as the thermal conductivity
- steam conditions, such as the outlet temperature and pressure, inlet temperature, and heat transfer coefficient (function of mass flow rate)

This analysis inherently involves the determination of appropriate materials (and location for material transitions) for different segments of the superheater/reheater sections when designing a steam generator, because the material properties and tube geometry (function of material properties) effect the overall heat transfer rate, which in turn effects the tube temperature.
A similar analysis, with GTL known and external heat transfer coefficient unknown, was performed on the Plant A steam loop to predict the spatial variation in tube and steam temperature.

6.4.1 Outline of GTLA Model

The GTLA model incorporates principles of heat transfer and fluid flow characteristics, specifics of which have been discussed elsewhere [398,400]. A brief outline of this model is presented here for context to the analysis performed on the Plant A AUSC steam loop. Note that the Plant A steam loop is a single tube circuit; thus, there is no need to account for differences in the external heat source or in the steam flow across the width of the steam generator, which would be necessary for GTLA of an entire superheater or reheater circuit.

The model is two dimensional; one dimension is through the tube wall, and the other dimension is along the tube length. The model is primarily applicable when the steam circuit is operating at a steady state; that is, the temperature of the tube metal or the steam temperature at any spatial location along the gas-touched length is not varying with time. Such a condition is attained when the heat flux through the tube is constant. Again, this application of a design tool is to gauge the effectiveness of the first-principles predictions of a dynamic analysis.

6.4.2 Development of GTLA Model

The radial distribution of temperature across the tube wall is shown in the schematic in Fig. 6.43 where the blue line represents the temperature profile from the gas (left) to the center of the tube (right).

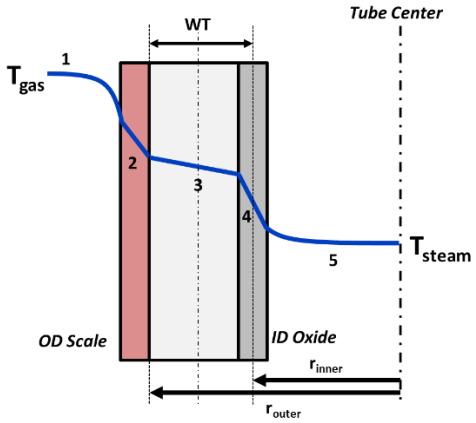


Fig. 6.43. Schematic diagram illustrating the distribution of temperature across the tube wall from the flue gas region (left) to the center of the tube (right).

The model includes conductive and convective heat transfer with fluid flow hydraulics. To determine the temperature at various boundaries the heat transfer coefficients are calculated for the regions marked 1, 2, 3, 4 and 5 in Fig. 6.43.

Region 1: Temperature drop due to fireside fouling (this can be included if the thermal resistivity due to fouling is known)

Region 2: Transition in temperature due scale formed on the OD surface, which is determined using conductive heat transfer across the scale thickness

Region 3: Change in temperature across the tube wall thickness due to conductive heat transfer through tube wall.

Region 4: Change in temperature due to conductive heat transfer across oxide scale formed on the ID surface

Region 5: Change in steam temperature from the tube ID to the center of the tube resulting from convective heat transfer in steam. Region 5 can also include a steam side fouling term.

The gas-touched length of a steam circuit can be discretized into small tube segments of length ΔL (as shown in Fig. 6.44) and the principles of heat transfer is applied to this small tube section. The cross-sectional plane 'A' represents the inlet and the cross-sectional plane 'B' is the outlet end of the tube section.

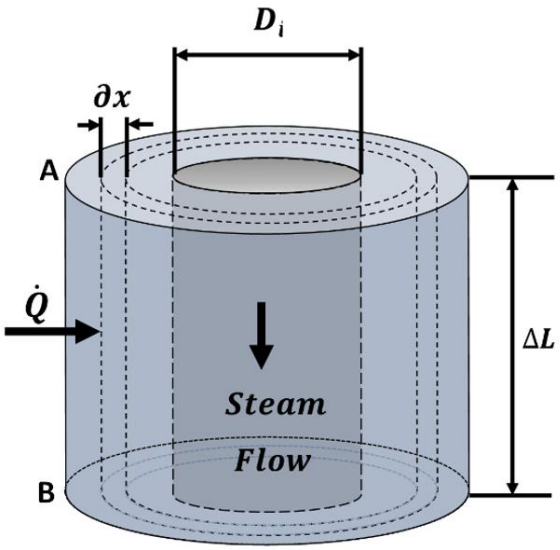


Fig. 6.44. Schematic showing a discretized tube element of length ΔL, where 'A' is the steam inlet plane and plane 'B' is the steam outlet plane. Radial distribution of temperature along plane 'A' of such a discretized element is calculated in the GTLA model.

The following necessary parameters were used as inputs for the model:

Operational Parameters:

- steam temperature at the tube inlet (T_i) [represented by the sectional plan 'A' in Fig. 6.43]
- steam velocity or steam flow rate ($\dot{m}$)
- pressure at tube inlet (P_i) [represented by the sectional plan 'A' in Fig. 6.43]
- temperature of Gas (T_g)

Material Parameters:

- thermal conductivity (κ)
- material dimensions (outer diameter, wall thickness, segment length for a grade)

Other Optional Parameters:

- thicknesses of scales formed on tube outer and inner surfaces [gas side scale and steamside oxide scale]
- thermal conductivity of scales

In addition, access to energy transfer properties of saturated and supersaturated steam (via steam tables) would be required.
Conductive and convective heat transfer equations are applied as applicable to regions mentioned above to determine the radial distribution of temperature. These equations are:

1. Conductive heat transfer through the wall is given by:

$$\frac{\partial Q}{\partial t} = -\kappa A \frac{\partial T}{\partial x} \tag{6.4}$$

Where:

- Q is heat that is transferred
- ∂Q/∂t or Q-dot is the heat transfer rate
- κ is thermal conductivity
- A is surface area across which heat is being transferred
- ∂T is the change in temperature over a distance of ∂x

2. Convective heat transfer due to the steam within the tube is described by [401]:

$$h = \frac{\kappa_{steam(T,P)} 0.023\ Re^{0.8} Pr^{0.4}}{D_i} \tag{6.5}$$

- h is convective heat transfer coefficient
- $\kappa_{steam(T,P)}$ thermal conductivity of steam at a temperature and pressure
- *Re* is Reynolds number
- *Pr* Prandtl number
- D_i diameter of orifice

Using Eq. (6.4)and (6.5) the radial distribution of temperature from the tube OD to the center of the tube at the inlet (sectional plane 'A' in Fig. 6.43) can be estimated.
For any given radial location of interest, to determine the variation in temperature along the tube length, a third equation is used, which calculates the drop in pressure of the steam along the length of the tube.

3. Drop in the pressure along a tube element of length ΔL is given by.

$$\Delta \boldsymbol{P} = \frac{\boldsymbol{8 f \dot{M}^2 \Delta L}}{\boldsymbol{\pi^2 \rho_T D_i^5}} \tag{6.6}$$

- ΔP is drop in pressure along a length ΔL of loop
- f is frictional factor (steam is assumed as viscous fluid)
- $\dot{M}$ is mass flow rate
- ρ_T density of steam at that temperature

By applying Eq.(6.6), the drop in steam pressure between the inlet (sectional planes ‘A’) and outlet (sectional plane ‘B’) of the tube element shown in Fig. 6.44. Schematic showing a discretized tube element of length ΔL, where ‘A’ is the steam inlet plane and plane ‘B’ is the steam outlet plane. Radial distribution of temperature along plane ‘A’ of such a discretized element is calculated in the GTLA model., can be estimated, which is then used to determine the steam temperature at the tube outlet (sectional plane ‘B’ in Fig. 6.44) using steam tables. This steam outlet temperature is then taken as the inlet temperature for another adjacent tube segment of length ΔL and the calculations are repeated for the new tube segment. Extending these calculations for the whole gas-touched length will provide the variation of temperature along the gas touched length of the loop at any chosen radial location. [For example, the temperature profile along the gas-touched length of any of the interfaces (represented by the points A, B, D and E) and the center of the tube (represented by the point C in the schematic of a transverse cross section of a tube in Fig. 6.45) can be estimated.]

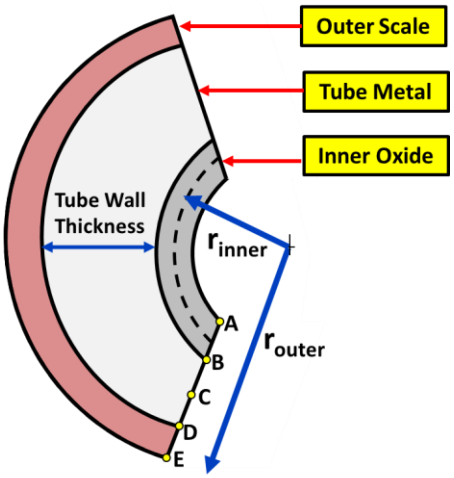


Fig. 6.45. Schematic of a transverse section of a tube wall showing various interfaces present in the radial direction (A, B, C, D and E) where the steady-state temperature can be calculated.

The GTLA model prepared as described above was applied to the AUSC steam loop under a variety of real-world conditions observed during the operation of the steam loop to explore the possible profiles resulting from high and low load, inlet temperature, etc. In Fig. 6.46, three such conditions are presented representing a combination of low and high flow and low and high delta temperature as described in Table 6-7. These values were selected from the operational data from the plant during the window where the flue gas thermocouples were operational and represent the impact of real conditions. This is a demonstration of the GTLA calculation and how it predicts different temperature profiles based on different conditions.

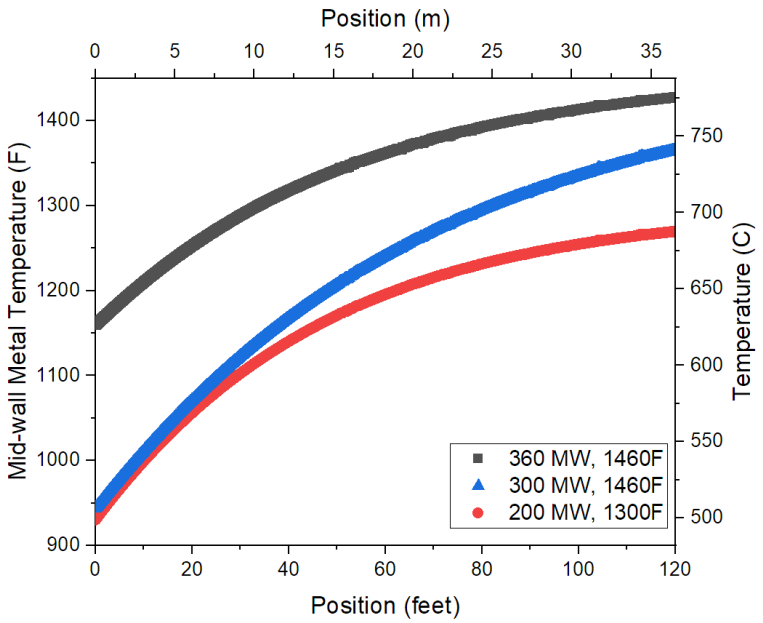


Fig. 6.46. Predicted temperature profile along the AUSC steam loop under conditions described in Table 6-7.

Table 6-7: Example operation parameters evaluated for GTLA of the AUSC Steam Loop representing the range of observed conditions.

Unit Output (MW)	Outlet Flow (lbs/h)	Inlet T (°F)	Pressure (psi)	Outlet T (°F)	Inlet Flow (lbs/h)	Delta T (F)	Ave Flue Gas T (°F)	HTC (calc) (Btu/hr-ft^2-F)
360	2577	990	572	1411	1176	422	1460	98.9
300	707	828	449	1347	219	519	1460	13.3
200	586	800	296	1260	167	460	1300	17.9

6.5 Thermal Conductivity Tests

6.5.1 Heat Transfer Sensitivity to Oxide Conditions

The thermal conductivity and heat conduction of metal oxides may play a significant role in the evolution of heat transfer in boiler tubes with time. This impact is especially important to the life management of the steam-touched superheater tubes which experience the highest average metal temperatures. Low alloy steels will grow oxides faster at a given temperature than higher alloy steels, e.g. T22 vs. T91 vs. stainless, which is a factor in how those alloys are selected in design. Metal temperature is the primary determining factor for oxide growth.

A unit operating dynamically (a.k.a. flexible operation or load following) can complicate temperature distributions in a boiler, reducing uniformity in space and time. Changes in the magnitude and location of peak temperature in the boiler can result in nonuniform oxide growth. As oxides get thicker, their contribution to resisting heat transfer increases, but due to the complex nature of such oxides, this behavior may not be linear over time nor with scale thickness. Oxides on ferritic steel tubing will form a multi-layered structure of various oxides (e.g., FeO, Fe_2O_3, Fe_3O_4) based on oxygen potential at the oxide metal interface. Some oxides are considered protective and desirable, while others are prone to cracking and delamination; this section focuses on the oxides formed in superheater and reheater tubing. Porosity within the oxide scale, cracking of the oxide scale, and partial exfoliation increase the varied conditions of the oxide along the length of the tube and across tubes though the influence of porosity, large pores, or high densities of pores do not demonstrate a clear trend on oxide properties or the potential for exfoliation [402]. Some of these effects may reduce heat transfer resistance – and reduce the local metal temperature – and as such are expected to reduce the accumulation of creep damage. The rate of regrowth after exfoliation may not follow the original kinetics which makes it more challenging to predict. Moreover, through-scale cracking can re-expose the underlying substrate to accelerated local attack leading to nodules of much thicker scale. All of these possibilities lead to a wide distribution of oxide scales throughout the system.

6.5.2 Existing Oxide Thermal Conductivity Values in Literature

Given that oxide contributions of metal temperature are important to the self-accelerating process of creep damage accumulation, it is important that the accumulation of damage to superheater tubing accurately account for the effect of oxides and uses relevant material parameters. While there have been some experiments performed and measurements taken on representative oxides from power plant boilers, Fig. 6.47 [400] there is a low agreement in the thermal conductivity values over a relevant temperature regime and when considering the influence of cracking, porosity, and other morphological phenomena.

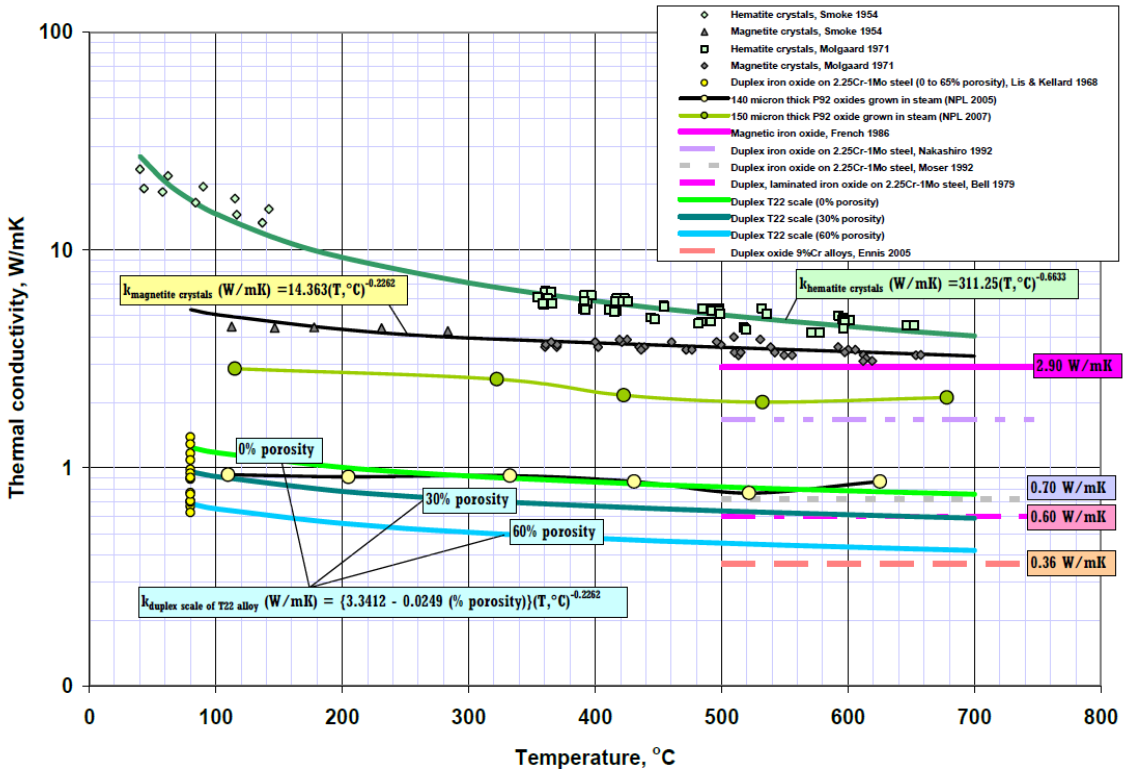


Fig. 6.47. Survey of thermal conductivity of metal oxides reported in the literature [400]

Attempting to refine this understanding was considered to be of significant value to this project and as such the team sought to evaluate ex-service oxide scales using modern techniques to seek to reduce uncertainty in the estimating the appropriate value of thermal conductivity of iron oxides.

6.5.3 Laser Flash Diffusivity Testing of Ex-Service Tubes

The project evaulated laboratories with expertise in measuring thermal conductivity regarding state of the art procedures and equipment for measuring thermal conductivity of multi-layered materials. Laser flash diffusivity (ASTM E1461), which projects an intense energy burst onto one side of a specimen while the temperature is measured from the other side, was selected for this analysis. While this test method can be applied to multi-layered structures, this raises several concerns. The highest fidelity in measurement is possible only when the contribution of each layer can be isolated individually with a high signal to noise ratio. This is best achieved by thinning one layer to allow for the overall contribution of the other layer(s) to be detectable. In the example of tube oxide, a bare metal tube can be tested and the contributions of that analytically removed during the analysis of the tube plus oxide multilayered sample to give a result for just the oxide. Further consideration must be afforded to any inter-layer thermal conductivity changes, such as may be present in porous or partially disbonded layers. The samples evaluated in the experimental studies represent a range of microstructures which helps to bound possible oxide thermal conductivity values.

The project completed laser flash diffusivity testing of five representative ex-service tubes. This task had the aim of refining our estimates for the thermal conductivity of metal oxides from literature values as oxide growth is considered a self-accelerating input to the damage evolution portion of a tube life model.

The team worked with an external vendor to perform ASTM E1461 testing at room and elevated temperatures. The laser flash method flashes one face of the specimen (in this instance, the metal face) with an intense laser energy burst. The radiative energy heats up the surface which then conducts that energy to the other face of the specimen. The dark, non-illuminated face is monitored with high frequency optical temperature sensors. From the time delay behavior of dark surface heating, thermal conductivity can be calculated. This approach allows for the evaluation of multi-layered structures, such as oxides on metal substrates, provided some values are known about the substrate. The thermal properties of tube metals are well known and were measured for these experiments allowing for the analysis of the conductivity of the surface oxides.

Measured values for thermophysical properties used in this analysis are listed in Table 6-8. Thermal diffusivity is the key value measured in a laser flash test. It is only by knowing density, specific heat, and diffusivity that thermal conductivity may be calculated. The laser flash testing of ex-service material produced a wide range of thermal conductivity measurements for what would be expected to be very similar steels. Values of thermal conductivity or bare metal specimens spanned a factor of 3 which is not expected nor supported by literature values. Thermal conductivity of the base metal is not expected to change significantly with temperature.

Given discrepancies between base metal properties in initial experiments and other literature sources, additional as-manufactured base metal samples were provided to the test vendor to verify the approach. These revised conductivity measurements are presented in Table 6-8. The outcome of using this revised metal thermal conductivity in the original experimentation had a small impact on oxide thermal conductivity, Table 6-9.

Table 6-8: Thermophysical properties of ex-service tube materials as determined by laser flash diffusivity testing by the external contractor. These thermophysical property values were used in the calculation of oxide thermal conductivity as determined by laser flash diffusivity and presented in Table 6-9.

	T22	**T91**	**T92**	**Oxide**
Density (g/cm^3)	7.83	7.78	7.83	5.24
Specific Heat, RT (cal/g-C)	0.106	0.115	0.104	0.137
Diffusivity (cm^2/s)	0.133	0.045	0.100	
Ex-service Conductivity (W/m-K)	46.3	16.7	34.2	1 - 30
As-manufactured Conductivity (W/m-K)	29-39	25-30	22-27	1 - 20

The resultant oxide thermal conductivity data from the laser diffusivity testing is listed in Table 6-9. Additional samples were prepared with machined outer diameters to increase the ratio of oxide to metal in the test coupon. It was expected that the increased ratio would improve the sensitivity of the experiment. The metal and oxide thicknesses are reported in the table. Due to the complex nature of the oxide, metallography was performed on each specimen after testing to help explore the microstructure-to-properties relationships, Fig. 6.48. A qualitative description of the stage of oxide growth [402]is included in the table as well:

- Stage 1: oxide has formed a simple duplex (magnetite/spinel) structure
- Stage 2: oxide shows a laminated structure beneath the duplex (T22) or Cr-rich layers in the spinel (T91)
- Stage 3: hematite forms on the exterior surface and porosity/voids are present at the oxide-layer interfaces
- Stage 4: cracking and scale exfoliation and/or delamination occurs, generally of the magnetite layer
- Stage 5: magnetite regrowth is observed where exfoliation had occurred

After preliminary assessment, the results presented in Table 6-9 do not correspond well with literature values. For direct comparison, see Fig. 6.49 where the newly generated laser flash data are overlayed on the previous literature data. Here the oxide scale thermal conductivity as measured by laser flash diffusivity on ex-service tube specimens is compared to the previously reported values, largely from in-steam and in-air lab-grown oxides.

While porosity and cracking are expected to play a role, and may increase thermal conductivity, the discrepancy and scatter observed between the new data and historical data is concerning. In contrast to the literature data presented in Fig. 6.47 (specifically [403]), the recent laser flash diffusivity tests on ex-service material, Fig. 6.49, indicates that, in T91, porosity in the oxide increases the thermal conductivity of the oxide. However, the recent testing on ex-service T92 (~10 W/m-K) has poor agreement with published T92 data (~2 W/m-K). It also bears mentioning that this laser flash diffusivity data shows significant scatter between measurements, even within the same tube. Sources for error could be related to the test and sample setup, the measurement and interpretation of the data, or inconsistencies in the oxide present be they delamination (thickness), cracking, porosity, and other factors. These inconsistencies may be representative of in-service tubing but complicate the interpretation of laboratory data and recommendations for future work.

Table 6-9*: A summary of laboratory laser flash diffusivity testing results on the ex-service tube specimens.*

Tube Material	ID	Machined Thickness (um)	Oxide Thickness (um)	Oxide Stage[1]	Oxide Effective Conductivity (W/mK)			Revised Oxide Effective Cond. (W/mK)		
					RT	600C	650C	RT	600C	650C
T22	D1-1 D1-2 D2-1 D2-2	980 1002 2017 1981	172 175 162 161	3[2]-4[4]	7.2 7.7 11.9 9.3	6.3 2.4	6.5	7.6 87.2 12.1	2.6	7.9
T91	A-0.5 A-1.0	412 832	232 253	3[2]	N/A			N/A		
	B1-0.5 B1-1.0 B1-2.0	345 661 1625	225 353 323	3[3]-4[4]	26.7 4.2	 1.8 6.9	13.1 3.8 5.3	14.5 3.3	 1.7 6.9	15.2 3.8 6.8
	C1 C2	402 192	100 108	3[2]	1.5 2.6	 2.9	2.6 3.2	1.2 2.1	 2.7	2.0 3.1
Super 304H	E1-0.5 E1-1.0 E2-0.5 E2-1.0 E2-2.0 E3-1 E4-1	581 1135 529 1174 1978 1982 2090	28 28 36 28 12 43 37	2	N/A			N/A		
T92	G1-1 G1-2 G2-1 G2-2	915 998 408 439	96 83 90 87	3[2]	6.6 14.1 5.0 4.3	7.6 7.5 5.7	9.5 5.9 11.0	17.4	15.9 10.5 7.9	19.6 7.9 17.3

[1]EPRI's Oxide Scale Handbook [7] defines five stages of oxide growth for ferritic steels.

[2]Qualitatively, these samples showed limited porosity/voids in the oxide (<5%) as compared to…

[3]Qualitatively, these samples showed very high porosity/voids in the oxide (>20% of the spinel layer).

[4]Cracking and exfoliation observed in these specimens may have been caused by machining, sample preparation, handling, or testing in addition to in-service degradation.

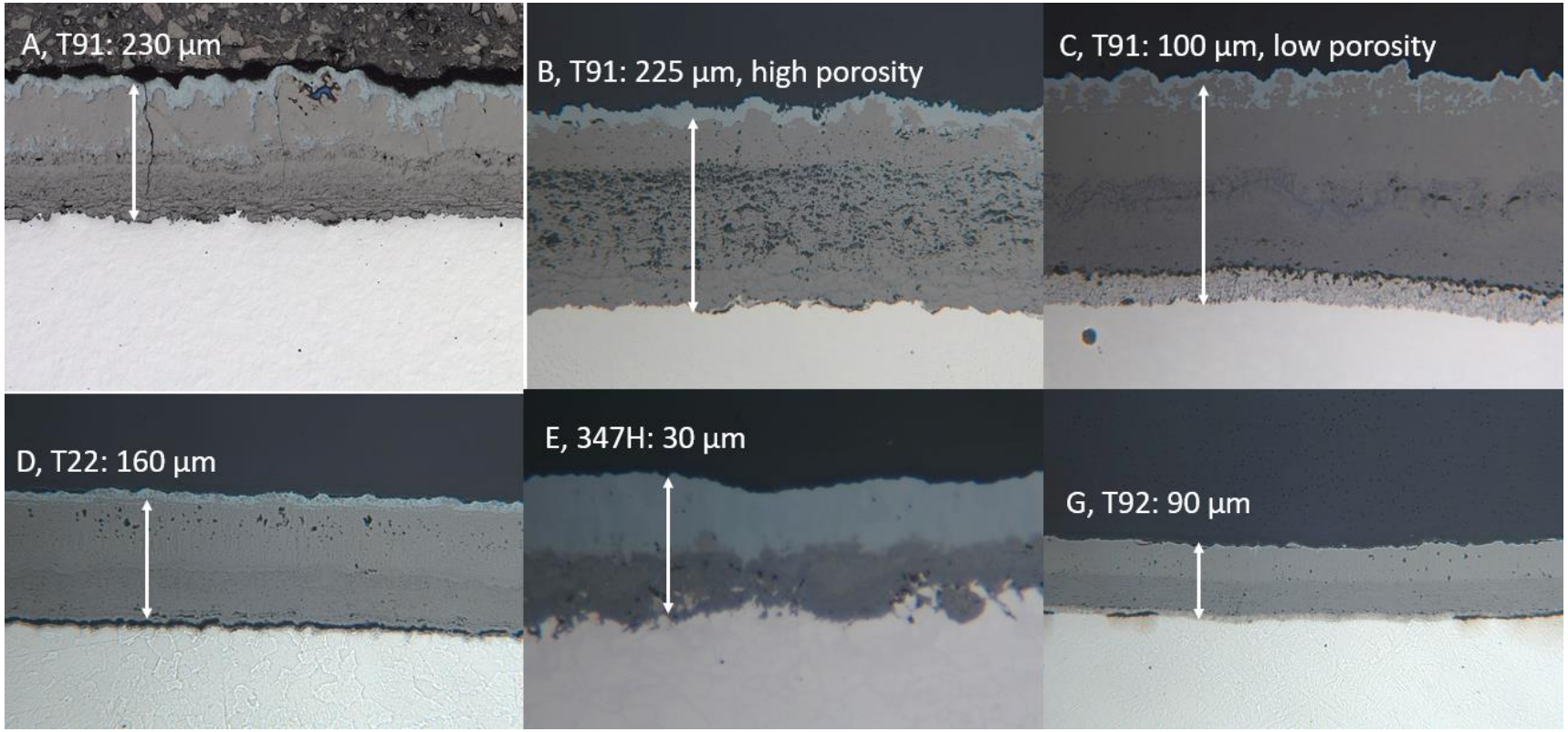


Fig. 6.48. *Representative optical metallography of ex-service tube samples A (Stage 3), B (Stage 3), C (Stage 3), D (Stage 3), E (Stage 2), and G (Stage 3). (These images are not all to the same scale.)*

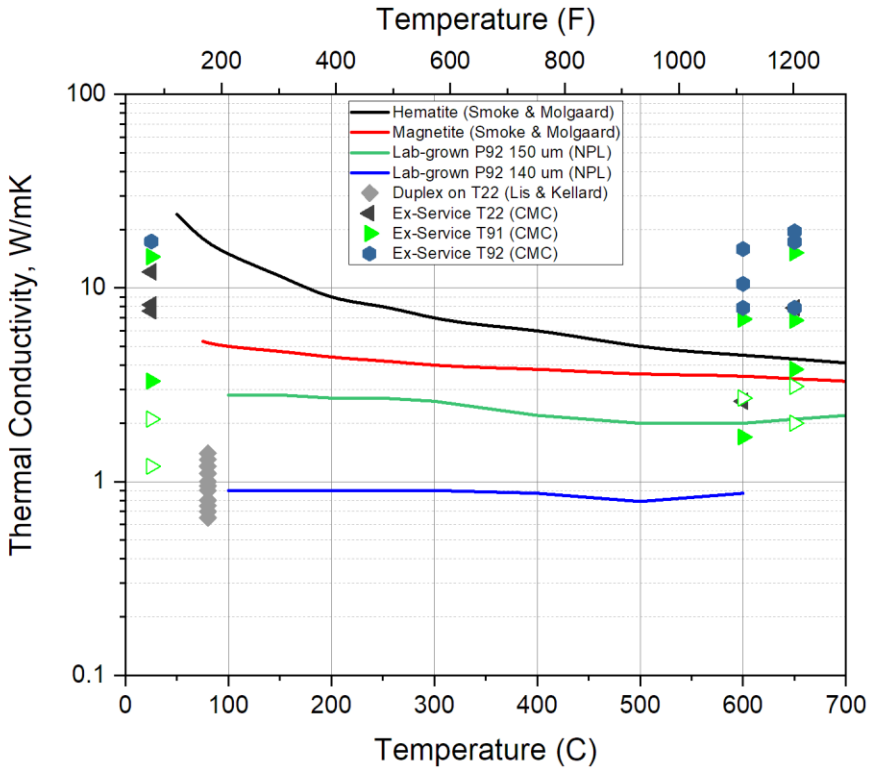


Fig. 6.49. Simplified Historical metal oxide thermal conductivity data with new test data (CMC) overlayed. T91 samples from "C", with a thinner lower porosity oxide are shown as open symbols and show a more consistent conductivity than thicker, more porous data.

6.5.4 Water Bath Heat Flux Experiment

As a second measure of thermal conductivity of metal oxides on tubing, the project team has also researched other laboratory setups that interrogate heat transfer across a larger surface area of an ex-service tube segment. This method may prove to yield results averaged over a larger sample (and distribution of oxide thicknesses) and could be an alternative approach to obtaining results relevant to assessing the performance of in-service tubes. Such an alternative method would also offer a way to validate or compare with the results from laser flash diffusivity. As shown in Fig. 6.47, thermal conductivity of oxides is only minimally affected by temperature; measuring most of the collection of ex-service samples at room temperature may offer an economic method to collect a volume of data.

This test method used an in-house water bath heat transfer setup, Fig. 6.50. This experiment is based on previous thermal experiments [400] with a heat source contained within a tube and a circulating water bath outside the tube. The experiment measures the water bath temperature increase relative to the heat input inside the tube which together can be used to calculate a heat flux or, as in this case, a resistance to heat flux caused by the tube and the oxide present. A further schematic of an expected temperature profile is show in Fig. 6.51. Experiments utilizing this setup have been completed on a variety of tube samples with a focus on demonstrating whether the setup can measure expected values, such as on pure metals, before proceeding to ex-service (oxidized) tubes.

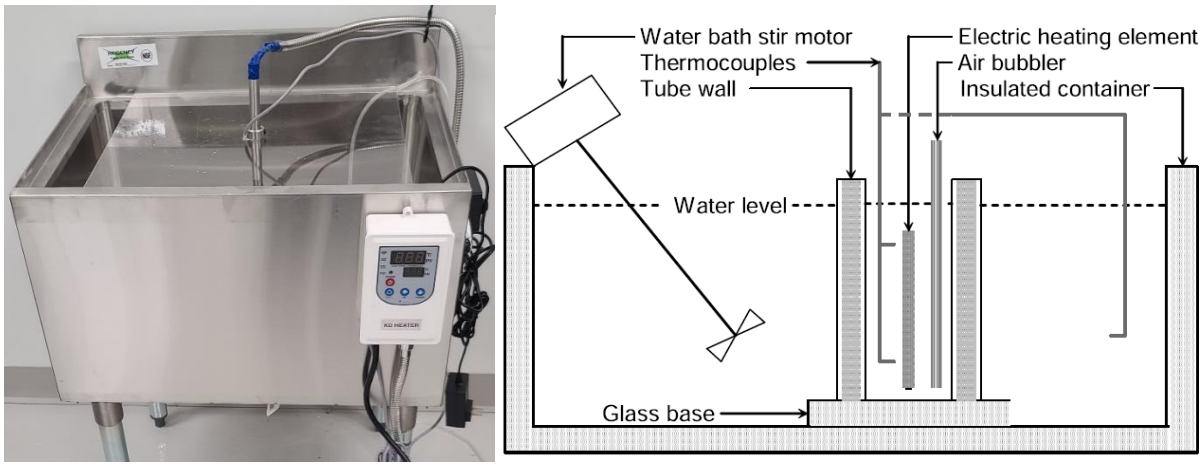


Fig. 6.50. Photograph (left) showing the experimental setup for the water bath heat flux testing at the EPRI laboratories. The associated schematic diagram (right) identifies key parts of the equipment.

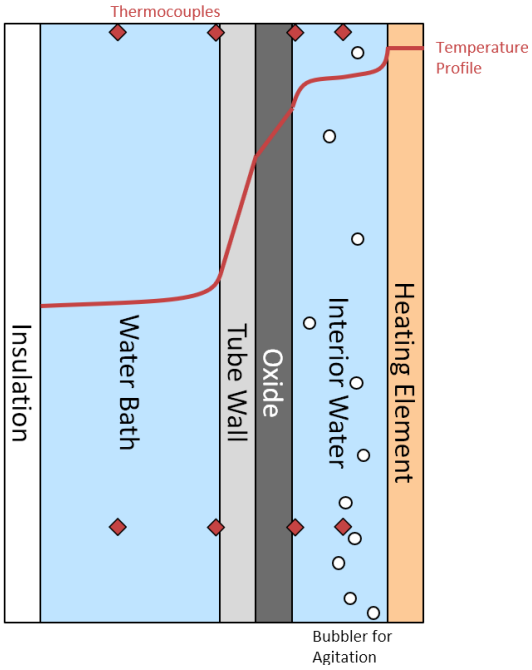


Fig. 6.51. Schematic representation of typical temperature profile illustrated across one half the apparatus which develops during the heat flux experiment. This figure also highlights the approximate location of attached thermocouples in EPRI's implementation.

The key equation for calculating thermal conductivity of an intermediate layer subject to a heat flux is reproduced here [400]:

$$k_{oxide} = \frac{R_o ln\left(\frac{R_i}{R_{ox}}\right)}{\frac{\Delta T}{\frac{Q}{A_o}} - \frac{1}{h_o} - \frac{R_o}{R_i h_i} - \frac{R_o ln\left(\frac{R_o}{R_i}\right)}{k_{metal}}} \quad (6.7)$$

where k represents thermal conductivities as labeled; R_o is the outer radius of a tube containing an internal oxide, R_i is the metal-oxide interface radius, and R_{ox} is the internal radius; h represents heat transfer coefficients at the locations identified; ΔT is the measured temperature differential between the inner and outer water baths; Q/A is the heat flux through the system; and:

$$h_i = h_o = \frac{R_o + R_i}{R_i\left(\frac{\Delta T}{\frac{Q}{A_0}} - \frac{R_o ln\left(\frac{R_o}{R_i}\right)}{k_{metal}}\right)} \quad (6.8)$$

where h is as determined by bare-tube testing in the same setup. For reference, the heat transfer coefficient calculated and used in the previous experiments [400]was 286.5 Btu/hr-ft^2 (1625 W/m^2-K) and the heat transfer coefficient calculated in this setup using a bare copper alloy tube (diameter 61 mm and 2.5 mm wall thickness) was 238 Btu/hr-ft^2-F (1350 W/m^2-K). These heat transfer values can be considered acceptably similar to each other for slow moving water-to-metal interactions.

Initial data has shown proof of concept both with the copper tube and ex-service ferritic steel tubing. The team has investigated the effect on results of using different small heating elements in order to provide the flexibility to have thermocouples located inside the tube during testing. TCs located too close to the heating element or are touching the wall of the tube will not accurately represent the inner water temperature. Six different tubes were tested using this setup as described in Table 6-10. The experiments used a fixed 9.5 L of water (2.5 gallons) and ran with a 200W cartridge heater. The experiment relies on constant heat input so the heater's controller was set to target 60°C so it would use its maximum heat input as it tried to reach that temperature – the experiments were interrupted well before reaching 60°C. The setup was first run with a copper alloy tube to develop a baseline of the heat transfer coefficient (HTC) between the agitated water and the metal tube. This HTC (1350 W/m^2-K) was assumed to be constant across all tubes' inner and outer surfaces; implying no impact on HTC due to the presence or absence of oxides. Each experiment produced datasets from eight thermocouples (as in Fig. 6.51) that capture the temperature at different locations on the tube and in the water, Fig. 6.52. This data was analyzed per the above equations to produce a thermal conductivity of the oxide layer for each specimen.

Table 6-10: Geometric characteristics of ex-service tubes tested in thermal flux experiments

Tube	**A**	**B**	**C**	**D**	**E**	**F**
Material	T91	T91	T91	T22	T347H	T92
OD (mm)	49	51	44	50	44	44
ID (mm)	41	33	30	27	28	31
WT (mm)	3.9	8.9	7.2	11	7.9	6.7
Oxide (um)	240	330	100	165	40	80

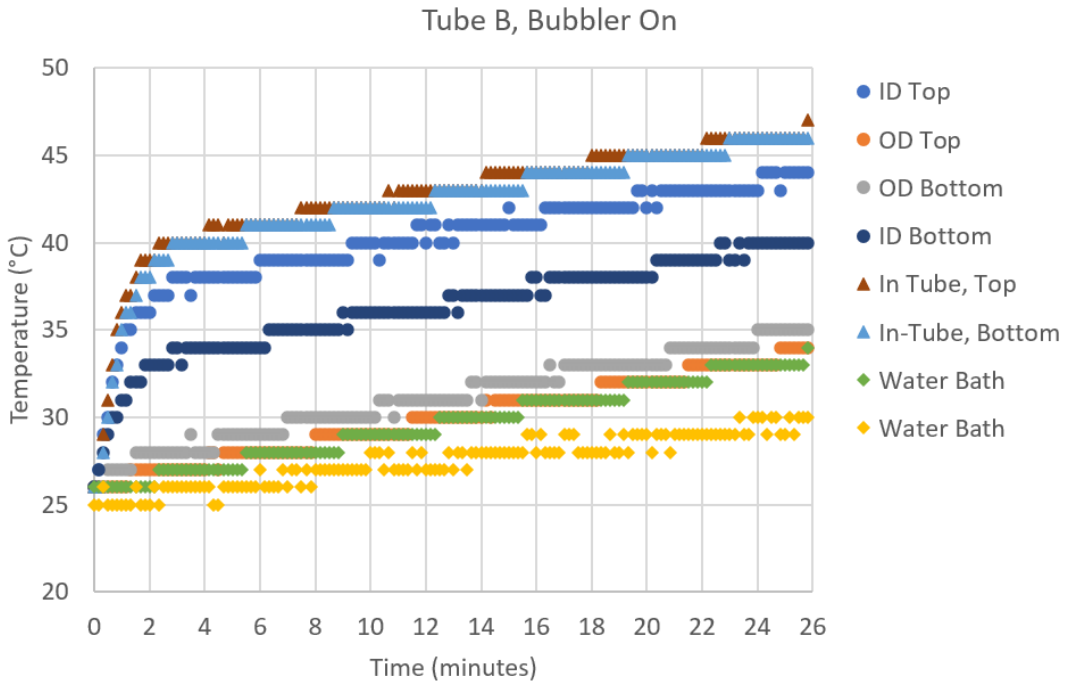


Fig. 6.52. Example plot of heat flux experiment data, this data from Tube B (with the bubbler on)

Fig. 6.52 shows the first results obtained for the T91 tube 'B' with the gas bubbler in use. This Figure shows the variation with time of temperatures measured from the eight thermocouples in the system. The internal water (triangles) reaches the highest temperatures facilitated by the capsule heater and bubbler circulation. Heat transferred through the oxide and into the metal (circles) where TCs measure a gradient through the wall. The through-wall gradient in this example is more sever at the top of the tube than the bottom. Finally, the water temperature (diamonds) showed a constant gradual increase in temperature indicating the experiment has reached a sufficiently steady state.

This behavior is consistent across all specimens. However, removing the internal agitation causes a more significant temperature gradient between the top and bottom of the tube. This internal water showed evidence of having not fully mixed and thus the tube experienced a nonuniform heat transfer through the wall. The experiment does develop a thermal gradient between the water inside the tube and that of the water bath outside the tube. However, it appears as though the air bubbler which is acting as an agitator inside the tube is significantly influencing the amount of measurable heat flux in the system, to the point of introducing apparent negative thermal conductivity contributions from the oxide scale. The average metal-oxide thermal conductivity of these tests without the bubbler and at the top of the tube is 0.20 W/m-K; with the bubbler on the average is calculated to be -0.17 W/m-K. Further, it should be noted that even the 0.2 W/m-k value reported is significantly less than expected based on historical data, ~1W/m-K.

Given how crowded these tubes were with instrumentation and the heater, the experiment was performed in the reverse direction: heating the external water bath and monitoring temperature increases inside the tube with the hope that this configuration can reduce the error and uncertainty in the measurements. Again, a key concern has been that the heating element itself is taking up a significant portion of the volume inside the tube which is restricting the internal circulation. The internal diameter for the ex-services tubes used here are 27 to 41 mm while the heater itself measured 12 mm. On the smallest tube, that leaves just 7 mm of radial distance for thermocouple wires, water, and air bubbles. It is anticipated that having more free-flow of internal water will assist with temperature uniformity. In the reverse condition, only a TC is in the ID of the tube which should allow for easier temperature equilibrium of the inner water. To

accomplish a rapid heating of the entire water bath, we have reduced the volume to approximately 10L.

Five tubes were tested in the reverse water bath experiment, Table 6-11, including two of the ex-service dutchmen from Plant A.

Table 6-11: Geometric characteristics of ex-service tubes tested in heat flux "reverse" water bath experiments

Tube	**C**	**E**	**F**	**99**	**189**
Material	T91	T347H	T92	T22	T22
OD (mm)	44	44	44	44	44
ID (mm)	30	28	31	23	23
WT (mm)	7.2	7.9	6.7	10.5	10.5
Oxide (μm)	100	40	80	110	180
Sustained ΔT (C)	28	15	13	18	14
dT/dt (C/minute)	1.8	4.2	3.0	2.4	2.7
Q/A_i	301	655	512	314	357
k (oxide) (W/m-K)	0.0057	0.0186	0.0262	0.0151	0.0424

For this modified setup, the heat transfer coefficient was again measured first to be $h_i = h_o = 121$ Btu/hr-ft^2 (686 W/m^2-K) as determined by a non-oxidized sample in this setup.
Again, these results appear to indicate that this setup lacks the sensitivity to accurately measure the thermal conductivity of the oxide. Previous testing of metal oxide thermal conductivity would suggest a room temperature thermal conductivity between 0.5 and 5 W/m-K; and, given this previous experience, the values < 0.1 W/m-K reported in this experiment do not appear reasonable. Of note, the original implementation of this test setup [400] utilized tubes with a metal to oxide thickness ratio of about 10:1. This project's ex-service samples are closer to 100:1 so experimental sensitivity may be providing sufficient noise to mask the calculation of the oxide thermal conductivity.

6.5.5 Recommended values for WVU model and beyond

In the absence of new, reliable data, the project defaulted its recommendations for thermal conductivity of the metal oxide to the industry standard values. Some conservativism is warranted for this prediction and the suggestion was made to use 0.8 W/m-K for the physics-based model predictions. WVU-model predictions of oxide thickness leveraging this value for oxide thermal conductivity have been shown to match the oxide thickness measurements, see Section 5.6.3

6.6 Enhanced Life Assessment Tools

The principal intention of this work scope is to identify a path forward for a software or tools which would serve the needs of the power generation industry and address a framework for data-driven risk assessment. The direction taken here involved soliciting feedback from industry experts and outlining a plan of action for future tool development. EPRI leveraged its existing relationships with members (power generating utilities and plant operators) to discuss the need for a future tool which would process operational data and use the outputs from the analysis to estimate or quantify risk and/or damage. The industry expert input was key to framing the objectives for a future software tool (to be developed outside of this project). The plan developed concerned itself with issues such as:

- Which plant types should be selected,
- What inputs are needed,
- What analysis processes and algorithms need to be developed,
- In what format should the outputs be generated and presented,
- What assumptions are in effect, and
- What future work is needed.

This project's work scope is anticipated to end at the development of this plan. Execution of said plan would be up to the industry to lead based on its research priorities and resources.

6.6.1 Existing Tools

Conventional life assessment tools for superheater tube life management are generally static calculators which are used at selected operating times. Each calculation involves one set of inputs which produces a prediction of accumulated damage and/or remaining life. These calculators rely on manual user inputs for the specifics of a unit or component (fuel, geometry, materials, run-time to date), a single user-provided outlet temperature, and a set of assumptions around boiler operation (heat flux). The tool then calculates a temperature profile through the tube which yields an estimation of damage based on appropriate material properties. Uncertainty in the input values are accounted through appropriate distribution functions. Use of Monte-Carlo type simulation provides estimated failure rates [404].

Using tools of this type often requires the unit experience at least one tube failure. Subsequent analysis of the failure provides a known starting point for the condition of the tubing and its material properties. With informed inputs and simulations, the tool can be useful in predicting the next failure. The analysis process typically must also consider questions such as: is this first failure indicative of infant mortality, or is damage widespread so that a potential cascade of tube failures is likely?

An example of the output from this type of analysis is shown in Fig. 6.53. In this example, predictions of the cumulative failure probability have been made for T91 tubing which was assumed to have operated for 200,000 hours. If the steam outlet temperature is assumed to be at 560, 570, or 580 °C, the time to first tube failures (0.1% assuming 1000 tubes in the superheater) changes significantly from about 59,000 hours (>6 years) to 25,000 hours (<3 years) to 11,000 hours (>1 years), respectively. It is also apparent that at the highest operating temperature, 580 °C, the rate of tube failures increases more rapidly than is predicted at 560 °C Results of this type are

frequently used to inform future maintenance plans.

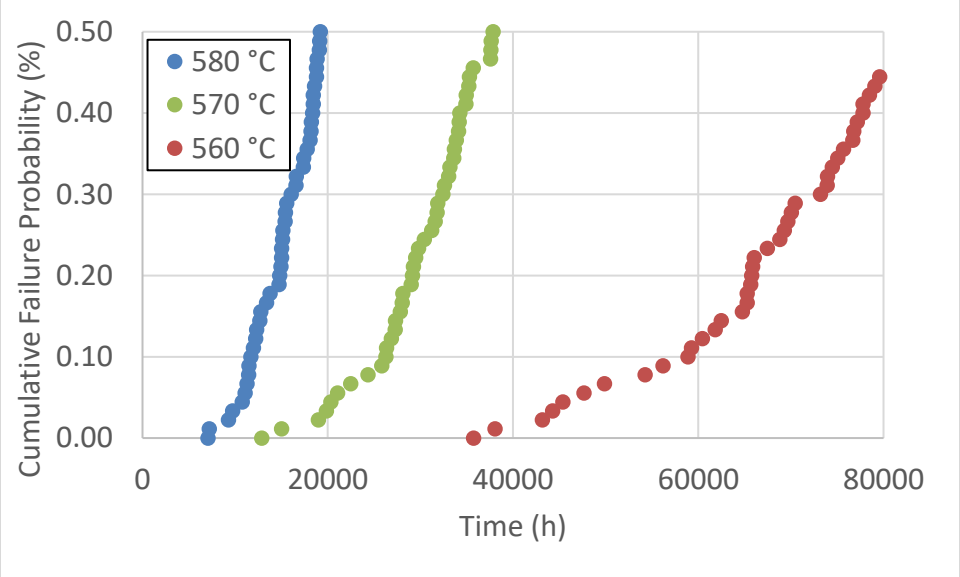


Fig. 6.53. Cumulative probability of failure versus time from a Monte Carlo simulation of T91 tubing under coal boiler superheater conditions after 200,000 hours of service, run by TULIP 2.0 [405]

Historically, this type of damage calculator has proven useful for reducing costs associated with the outages associated with boiler tube failures on units that generally run under base load conditions. As units spend more time away from full load and operating flexibly, the metal temperature distribution within the boiler (or specifically in sections of the steam superheater) becomes increasingly unpredictable. When a user runs the tube life tool under these circumstances, they may be making nonconservative assumptions on tube temperature leading to nonconservative predictions on life.

6.6.2 Survey of EPRI Members and their Future Needs

Traditionally the most efficient fossil fueled power generating plants operated as base load supplying electricity at a relatively steady level. Less efficient or relatively small sized combustion turbines were then used to generate electricity during periods of high demand. As the number of facilities producing electricity with no greenhouse gas emissions has increased, the operational characteristics of coal and natural gas burning stations has changed. In general, coal fired stations operate with complex patterns which involve more cycles to meet the changes in demand for electricity. The cycles experienced may involve stopping generation for a specific period (typically known as shutdowns) or operating at a low level of power generation. These events introduce challenges for the operators and associated engineering staff because the components within the generating stations experience changes in temperature and pressure. These variations will modify the local stresses experienced by components within the boiler and these changes can increase the rate of damage accumulation. Furthermore, the locations which are most at risk during cyclic operation can be different from those experienced under base loading. One key driver for the current project was to establish advanced analytical tools which would allow prediction of the temperatures experienced by the superheater sections of a boiler under all operating conditions.

EPRI staff interacts regularly with the engineering staff from a broad range of power generating companies. These interactions involve technology transfer associated with current approaches for operating stations safely, reliably, and efficiently as well as discussions of challenges and solutions. To provide commentary with respect to the specific analytical tools developed within the current project, a technology 'needs' questionnaire was developed. In all there were 21 questions which were designed to allow the maximum information to be gathered from the companies without necessitating unreasonable levels of effort. The questionnaire was distributed to staff in companies which have an established relationship with EPRI. In all cases, the companies receiving the questionnaire and their responses are confidential. However, the feedback provides valuable insight into current and future needs. The sections below summarize the responses for each of the questions.

1. **Do you have greater need for creep or borehole damage calculations in Coal fired Boilers or HRSGs?**

   The consensus was that creep damage is expected to be the mechanism of greatest concern. Because in-service experience with Heat Recover Steam Generators (HRSGs) is less extensive than that for coal fired units, HRSGs are higher priority. This ranking was reenforced by comments which indicate that many coal units have been assigned dates for closure. As a follow up comment, it was observed that in part because of the difficulties in performing inspections on many of the current HRSGs, the ability to accurately predict the locations and extent of damage would be a significant benefit to operators.

2. **Do you have greater need for assessment of superheater tubing or the performance of outlet headers?**

   Many responders indicated that there was an equal need for assessment capabilities for tubing and for headers. Differing geometries and materials (collectively, designs) complicate these assessments across the fleet. The challenge associated with assessment was also highlighted by the fact that in some cases the headers and associated fittings were fabricated with longitudinal seam welds. In general, the presence of seam welds in components operating in the creep range increases the risk of catastrophic fracture. There was a recognition that with more complex operating scenarios and greater variety of fuels achieving reliable performance would be more difficult. Thus, validated tools identifying the locations at higher risk of in-service damage would be useful.

3. **For superheater tubing which of these three options best describes your current practice?**

   - **Reactive / schedule maintenance**
   - **Condition Based through Inspection**
   - **Condition Based through Inspection + Data Informed**

   Most responders identified that inspection was the approach used for assessing the condition of tubing; however, it appears that at least for coal fired boilers, reactive approaches are becoming more common. Comments suggest that this change has come about because of reduced budgets.

4. **For superheater tubing how do you track damage between inspections?**

   The most frequent response was that there were no actions taken to track damage between inspections. One company indicates that a manual review of available temperature data is undertaken to ensure that the defined limit is not exceeded. Another indicated that a

commercially available damage estimation software package is being trialed.

5. **For superheater tubing how do you track damage at inspections?**

   Most responses indicate that nondestructive testing is undertaken, namely visual inspection and measurements of tube wall thickness and the thickness of steam side scale. Typically, the results were stored in spreadsheets. Several organizations identified that they were using a commercial package for storage and retrieval of plant inspection data. When dissimilar metal welds were present in the superheater (i.e. welds joining ferritic and austenitic steels), these were inspected using an established dye penetrant method.

6. **Does the damage tracking process include the use of installed thermocouples on the tubes?**

   The main response was that the output from thermocouples is not used for damage tracking. Even when there were thermocouples installed on the superheater tubing, these were generally attached to the outside of the tube in the penthouse, i.e. on the tube before the superheater outlet header. Of course, this is necessitated by the fact that attempting to use thermocouples where flue gas is present is problematic in terms of both the reliability and accuracy of the results. It was noted that where thermocouples were present, they were typically used as an alarm system so that operators could identify off normal conditions. One organization identified that a trial was on-going using wireless data transmission from installed thermocouples. The success of wireless transmission offers the potential for cost reduction and thus may increase the use of instrumentation in boiler applications.

7. **If you answered yes to Q6, what analysis do you undertake that makes use of the TC data?**

   Even in limited case where thermocouples are present, there was very limited analysis. Data recorded was manually checked against temperature limits. One plant was using temperature readings with a commercially available software product to estimate damage accumulation.

8. **Why do you consider TCs and data produced not worthwhile?**

   Tight budgets prevent widespread installation of additional thermocouples/instrumentation. Further, there are costs associated with maintaining this instrumentation. There is also a lack of confidence in model accuracy of predicting current component conditions based only upon thermocouple (or other historian) data. Based on previous experience, the stability of software implementing these models has been poor. Thermocouples have been used when there was a need for specific data related to a known component problem, e.g. thermal shock causing cracking.

9. **What percentage of your outages / lack of availability is due to failures of SH tubes?**

   Failures of superheater tubing do not currently cause significant forced outages. Some examples of cracking at weld between tube and header have been recorded.

10. **Do you know the cost (lost production and/or repair cost) of the failures over recent years?**

    For some, limited forced outages due to superheater tube failures prevents an accurate estimation of costs. At times when electricity prices are high a forced outage can cost more than $1 million.

11. **What are the biggest challenges to reducing these costs?**

Challenges include shortage of skilled labor, cost of replacement parts, and making high quality repairs.

12. **What software, if any, do you currently use to analyze data to support SH tube/header maintenance?**

    Most responses indicated that no software is used to support maintenance of SH tubes and secondary superheater outlet headers. There is some use of spreadsheets, and one unit is trialing a commercial software product.

13. **For superheater headers which of these three options best describes your current practice?**

    - **Reactive / schedule maintenance**
    - **Condition Based through Inspection**
    - **Condition Based through Inspection + Data Informed**

    Most responses indicated that condition was assessed using inspections. In one case the scheduling of the inspections was linked to operating time. In another, the schedule was based of analysis of plant data. However, it was also noted that in many cases utility staff with the necessary expertise were retiring and not being replaced.

14. **For superheater headers do you seek to track damage for the whole header?**

    Comprehensive inspections of potential damage mechanisms are performed periodically. More frequent inspections are performed for high-risk locations or areas with known damage. In some cases, a maximum time between inspections is defined based on previous experience. There are challenges in getting access for inspections of headers in HRSGs.

15. **Have you undertaken an evaluation of the most common SH header damage mechanisms and the likely susceptibility for your equipment?**

    Most respondents indicated that they are aware of typical damage mechanisms and plan component inspections accordingly. There is also an appreciation that cyclic operation is increasing the likelihood of damage developing due to fatigue.

16. **Do you use installed thermocouples to monitors temperatures? Do you consider your number of thermocouples:**

    - **Minimal**
    - **Typical**
    - **Comprehensive**

    Most responses indicated that although there are thermocouples on some tubes, many components operated without installed thermocouples. In some cases, thermocouples were installed according to the minimum requirements of the original equipment manufacturer.

17. **What analysis do you undertake to make use of the TC data?**

    Since many generating companies did not have thermocouples installed on headers, the question about types of analysis was not meaningful. In some cases, the temperatures recorded from the tubes were taken as being representative of the metal temperatures experienced by the headers. The data from the thermocouples is increasingly being reviewed to estimate temperature changes due to changing ramp rates.

18. **Have you experienced cracking on superheater outlet headers? Is the mechanism due to:**
    - **Creep**
    - **Fatigue**
    - **Not sure**

    The responses to this question were varied. Thus, while no cracking was reported in about 30% of responses, damage associated with creep and/or fatigue were reported by about 70% of the respondents. This variation is not unexpected given the diversity in age, unit type, and type of generating mode seen in the U.S. power generation fleet. However, it is perhaps noteworthy that even in cases where cracking has been recorded, it is not apparent that broad programs of monitoring and analysis are considered of value.
19. **What were the costs (lost production and/or repair cost) associated with this cracking?**

    Most responders indicated that forced outages due to damage in major components such as a header cost more than $1 million. Repeated outages from tube failures can also result in significant costs.
20. **What are the biggest challenges to reducing these costs?**

    There was some variation in the responses. Of particular concern were the challenges associated with accurately predicting the location and rate of damage development. Of course, making accurate predictions of this type does require knowledge of the local operating conditions. It was also noted that there are difficulties related to lack of knowledge of the through wall temperature gradients and concerns regarding component relevant materials properties. Performing nondestructive inspections requires significant cost; the challenges linked to inspecting at the right location with the optimum method at the most appropriate time are not inconsiderable. When inspections do not detect defects, there can be an increased reluctance to continue to inspect. It should be noted that EPRI-recommended best practice regarding asset management involves application of an integrated approach to evaluation.
21. **What software, if any, do you currently use to analyze data to support SH header maintenance?**

    The vast majority of responses indicated that no software is used to support maintenance planning.

Concluding Comments

Most of the feedback indicated that coal fired units will be closed and so the current approach is not to use Operating and Maintenance budgets to prevent damage and cracking. That said, the majority of components considered by the questionnaire would not be considered safety critical so the consequence of failure would be loss of unit availability rather than injury to personnel. Several of the responses identified that the number of expert staff with generating companies has reduced so that there is a reduced capability for power generators to include significant numbers of in-house experts. Thus, there was a position that while data capture and analysis were desirable, EPRI should strive to ensure that this process did not require significant staff resource.

6.6.3 Proposal Framework for Superheater Damage Evaluation Software

Stakeholder Needs

The development of the Superheater Damage Evaluation Software is driven by specific needs identified through discussions with key stakeholders, including power plant operators and maintenance teams. These needs shape the system's functionality, ensuring effectiveness and user acceptance. This section details the core requirements and expectations expressed by the stakeholders.

Reactivity vs. Proactivity

Current Practices: Stakeholders primarily rely on inspections to assess the state of headers and tubing. As coal boilers age and reach end of life, the focus has shifted towards a more reactive approach, rather than spending O&M budgets on damage prevention. In contrast, HRSGs have lower in-service experience and present inspection challenges due to limited access, which makes it difficult to predict where failures may occur.

Desired Change: There is a desire for more, simple tools that help users estimate damage and understand the associated operational risk to specific hardware components. For HRSGs, there is a desire for tools that will aid in locating areas of risk in superheater components, especially where inspections are difficult to perform.

Data Management Needs

Current Data Availability: Many stakeholders primarily base maintenance decisions on inspection results and other non-destructive testing methods. While inspection results are stored, damage is not tracked between inspections. Installed thermocouples are rare, and when available, they are not used to assess component condition.

Basic vs. Advanced Analysis: Stakeholders prefer basic, standard analyses over complex, real-time analytics. For example, they highlighted the need to identify high-temperature excursions likely to cause significant damage, focusing on practical, immediate insights that guide maintenance decisions. Any third-party tools need to account for protecting data and maintain user privacy.

Tool Functionality and Accessibility

Web-Based Solutions: There is a moderate interest in adopting web-based tools for broader access and easier data management. This includes functionalities where users can simply upload data to the tool, moving away from traditional, manual systems.

Data Upload and Analysis: Stakeholders envision a system where data can be easily uploaded and automatically analyzed to determine the appropriate analysis. This includes the ability to map tags and specify units, simplifying the transition from data collection to analysis.

Flexible and Adaptive Analysis Model

Core Model Concept: Stakeholders are interested in a core "model" within the system that performs different analyses based on uploaded data. This model should adapt its analyses based on the quantity and type of data provided. For example, if the software has previously been used to analyze a superheater and new sensor data becomes available, the software should be able to extend any analyses to those new sensors.

Suggestive Analytics: Stakeholders indicated that the number of subject matter experts within their companies has been trending downward. The software should clearly inform the user of what data is required for each analysis to ensure meaningful insights. The software should be straightforward and simple to use, minimizing the time and effort from staff to operate it.

User Interface and Result Display

Functional Interface: The system should feature an interface that allows users to easily upload

files and see what analyses can be performed with the available data.
Results Display: After analysis selection, the GUI guides the user through mapping their data and defining their superheater system. Stakeholders expressed the need for a "display results" template that consistently shows analysis results in a user-friendly format.
These identified needs underline the importance of developing a system that is technically capable and user centered. The Superheater Damage Evaluation Software aims to meet these needs by being simple to use, providing meaningful damage risk assessments, and aiding in localizing those risks to specific areas for further inspections.

Software Overview

The software will be engineered to evaluate creep and fatigue damage in coal boiler and HRSG superheater systems from sensor data uploaded by users. It will offer a suite of functionalities, guiding users through identifying their analysis options, mapping data channels to a three-dimensional superheater model, performing detailed data analyses, and generating insightful reports. These reports should include identifying timeframes where risk of creep and fatigue damage were particularly high, tabulating fatigue cycles at different stress levels, and tabulating creep life consumption by identifying time spent at high temperature and high stresses. In the GUI, these results will be used to highlight specific time frames in time series plots where operation led to high creep and fatigue life consumption and highlight locations where damage risk was high on a three-dimensional representation of the superheater. In addition to assessing creep and fatigue life, the software is expected to also estimate oxide growth and perform failure rate analyses of header stubs.
Before any analysis is performed, the software should help the user identify what data is needed to perform each analysis. It should provide a simple interface that allows the user to select what data they have available and display what analyses the tool can perform with that data. Alternatively, the interface should allow the user to select a specific analysis that they are interested in and display the specific data that is required to perform that analysis. The user should be able to launch a new analysis from this interface.
Once a dataset is chosen, the software should identify each column in the data file and allow users to define various settings and thresholds defining the pre-processing checks for their superheater system. Once an analysis is chosen and data is uploaded, the software should pre-process the data to rectify errors, such as rate of change checks and stuck value checks, and ensure consistency with previously existing datasets through statistical analyses. These analyses will verify that the distributions of new data align with historical data, ensuring reliable comparisons and assessments. Upon uploading time series data, users should also be able to input geometric information and material properties essential for the selected analyses.
The software's GUI should enable users to map all uploaded sensors onto a three-dimensional representation of the superheater system, enhancing the visualization of analysis results. For basic risk assessment analyses, the software should be able to aggregate load, temperature, flow rate, and pressure data over fixed time intervals (e.g., hourly). This aggregated data will be processed using a cycle counting algorithm to identify recurring fatigue cycles. It will evaluate periods of high-temperature operation under significant stress to assess high-risk periods concerning creep life. The results will be tabulated for the user, and the superheater representation will be color-coded to highlight regions exposed to the highest risk.
For advanced creep and fatigue assessments, users will be able to provide detailed geometry and

material properties of the superheater. This additional information will allow the software to calculate the specific stresses encountered, resulting in a more accurate assessment of consumed creep and fatigue life. Similar to the basic assessments, the software is designed to tabulate the magnitude and frequency of fatigue cycles and the duration spent at high temperatures under substantial stresses. It will display a color-coded graphical representation of the superheater to facilitate user comprehension of the results.

In addition to assessing creep and fatigue life, the software is designed to estimate oxide growth over time using the supplied time series data, leveraging an existing EPRI oxide thickness calculator. The user should also be able to upload data regarding header stub failure rates, along with the previously discussed time series data, to perform a Weibull analysis to predict when the rate of stub failures will increase significantly. This analysis will consider both the operational time to failure and the operating conditions in the Weibull regression.

The software generates outputs that help users identify specific regions of their superheater headers and tubing experiencing increased damage based on operational history. It calculates relative damage observed for each mechanism analyzed and for each specific hardware component that has been mapped. These outputs enable users to understand the primary damage risks associated with their superheater operations and identify where headers and tubing are incurring the most damage. Users can incorporate these insights into their asset management systems to optimize maintenance planning for their superheater systems.

Software Requirements

The EPRI Superheater Damage Evaluation Software framework is intended to be developed as a web application designed to be hosted on EPRI servers. It will feature a web-based interface that allows users to upload data, define features for their coal boiler or heat recovery steam generator (HRSG) system, and view analysis that informs the user of operation that has been particularly harmful to the superheater in terms of creep and fatigue life. The software will have three core functional areas: Data Access, Local Storage, and an Analysis Engine, each crucial for analyzing the operation of their superheater hardware.

The Data Access area defines how users are authenticated, ensuring they only have access to their own datasets. It also defines how data files are uploaded to the software and how analysis reports are made available to the user. Data files are expected to follow a clear, predictable format with column names that are easily digestible. Column names should clearly identify the measurement, units of measure, and location of the measurement in a standard format such that that software can easily parse the column name. The locations will be used within the software to map each measurement to a three-dimensional representation of the superheater system.

The Local Storage area defines the database and file structure that will be used to store all of the user data. Superheater system metadata, such as superheater geometry and material properties, will be stored in a series of SQLite (or similar) databases for fast and flexible querying and to keep overall storage requirements low. Time series data will be stored using InfluxDB, a database type that is optimized for storing time series data. Information about each data channel stored in the SQLite databases will be used to identify specific data channels in the InfluxDB for each superheater.

The Analysis Engine area defines how data is moved from data storage, scheduled for analysis, cleaned through preprocessing tools, analyzed according to user inputs, and displayed. The software will perform several data checks during preprocessing: range checks, rate of change

checks, stuck value checks, units of measure checks, and missing value checks. These checks are critical to ensure that the analysis performed by the software is accurate. A report of this data cleaning analysis should be provided to the user so they may adjust accordingly. Additionally, for repeat analyses for a particular superheater, the software is designed to compare the uploaded data to stored data to ensure that the new data is complete and does not show a statistically significant difference compared to any prior data. The software will provide the user with the functionality to define appropriate values and thresholds for these pre-processing checks, which will ensure that the software remains flexible to any superheater system. Finally, the Analysis Engine is also responsible for displaying results in an easy-to-digest manner.

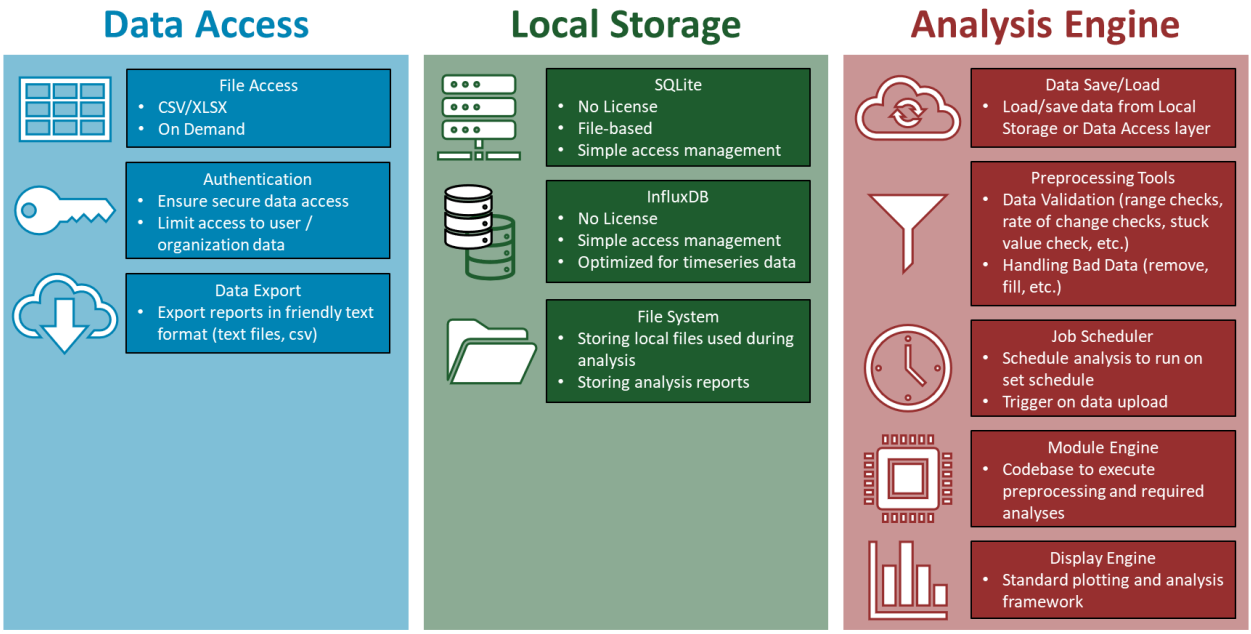


Fig. 6.54. Schematic of major building blocks associated with the development of software like the tool described in **Appendix D**

Analysis Requirements

The software is designed to provide a summary of creep and fatigue damage based on uploaded superheater data. At the most basic level, this includes analyzing load and steam temperature, pressure, and flow rate data to identify periods of high stress cycling and high temperature duration. With additional information, the software can extend these assessments to include stress calculations. Aside from creep and fatigue damage assessments, the software can also provide estimates of oxide growth and header stub failure rates. Specific analyses to be performed by the software are defined as follows:

"Risky" Operation Analysis: The software will analyze data on temperature, load, pressure, and flow rate to summarize stress cycles and the time spent at extreme temperatures. It will identify startup, shutdown, normal load fluctuations, and extreme ramp rates from load data. The software will also assess steam temperatures at the header and tubes for extreme values and fluctuations during startup, shutdown, and load changes. Additionally, it will quantify the number, magnitude, and frequency of cycles between different states of load, temperature, pressure, and flow rate.

Cycle Counting: The software should use a cycle counting algorithm, such as the range-pair or rainflow method, to identify different stress cycles in the data.

Fatigue Cycle Analysis: Adding geometric data and material properties enables the software to calculate hardware stresses to assess fatigue damage. The software should apply a cycle counting algorithm to these calculated stresses to tabulate the number of cycles with different stress magnitude changes.
Creep Life Analysis: Adding geometric data and material properties enables the software to calculate the time spent at high temperatures under high stress conditions to assess creep damage. The software should summarize this time to provide a comprehensive view of creep life consumed.
Oxidation Analysis: The software will use an existing EPRI calculator, either through direct integration or by retooling it, to estimate oxide growth. This estimation will give users a benchmark to compare against their own measurements.

Header Stub Failure Rate Analysis: Given data on header stub failures, the time of failure, and other related data channels, the software will perform a Weibull analysis to estimate stub failure rates over the superheater's lifespan.

Data and Metadata Requirements

To perform the defined analyses, the software requires several different types of data. These are defined as follows:
Time Series Data: Steam pressure, steam mass flow rate, plant load, and at least one steam temperature are required to analyze "risky" operations experienced by the boiler. Additionally, the number of tube assemblies and steam tubes per assembly are necessary to scale the steam mass flow rate to an individual steam tube, summarizing the "risky" operations experienced by each tube.
Geometric Data: Incorporating geometric data allows the software to calculate system stresses. Headers and steam tubes both require inner and outer diameters, wall thickness, and length.
Material Properties: Incorporating material properties, in addition to geometric data, extends stress analyses to include relative creep and fatigue life consumption. For fatigue life analysis, the modulus of elasticity, coefficient of thermal expansion, and Poisson’s ratio are required. Additionally, the Larson-Miller parameter is needed for creep life analysis.
Header Stub Failure Rates: At a minimum, age at failure and time since last maintenance are required for a basic Weibull analysis of stub failure rates. Uploaded time series data is also included to refine failure rates based on the operational profile, not just operational time.

Reporting Analysis Results

The software summarized the results of each analysis in the software’s interface. In additional to providing a tabular summary of each analysis, the interface also helps the user understand when operational risk was highest by highlighting specific regions in time series data and identifying specific regions in a three-dimensional representation of the boiler system. The specific results of each analysis are detailed as follows:
Startup Assessment: Provide summaries for cold, warm, hot, and fast startups detailing changes in temperature, pressure, and flow during these events, establishing baseline changes and identifying startups significantly faster than the baseline.
Shutdown Assessment: The software should summarize changes in temperature, pressure, and

flow during shutdowns, identifying shutdowns that deviate significantly from the baseline.
Transient Assessment: The software should include summaries of rapid changes in temperature, pressure, flow, and load data, highlighting any notable deviations
General Assessment: The software should detail the frequency and duration of load cycling, time spent at high temperatures, and load cycles occurring at high temperatures, providing an overview of the operational stresses.
Fatigue Cycle Analysis: The software should summarize the number of stress cycles encountered by the superheater at various stress magnitudes, offering insights into the fatigue damage observed.
Creep Life Analysis: The software will provide a summary of the time spent at elevated temperatures and stress magnitudes, which are critical factors in assessing the remaining life and risk of creep damage.
Oxidation Analysis: The software should include an estimation of oxide thickness growth over time.
Header Stub Failure Rate Analysis: The software should present summaries of header stub failure rates, including predictions for when failure counts are expected to rise rapidly, based on historical data.

Additional Functionalities

In addition to performing analyses that allow users to assess boiler operational risk with regards to creep and fatigue life, the software also provides functionality to help users understand what data is needed for each analysis. A tool has been designed that allows users to select data, geometry, and material properties to which they have access, and the tool will filter this list of analyses appropriately to inform the user of what analyses the tool can perform with that information.
The tool can also turn this information around to inform users of where additional data can provide additional analytical value. The tool also has an option to select the desired analysis, whereby it will filter the data, geometry, and material properties required for that analysis. This feature is particularly useful for users who are considering installing additional sensors, informing them of specific measures that directly improve their ability to assess creep and fatigue life consumption in their boiler systems. An example mockup of these functionalities is shown below.

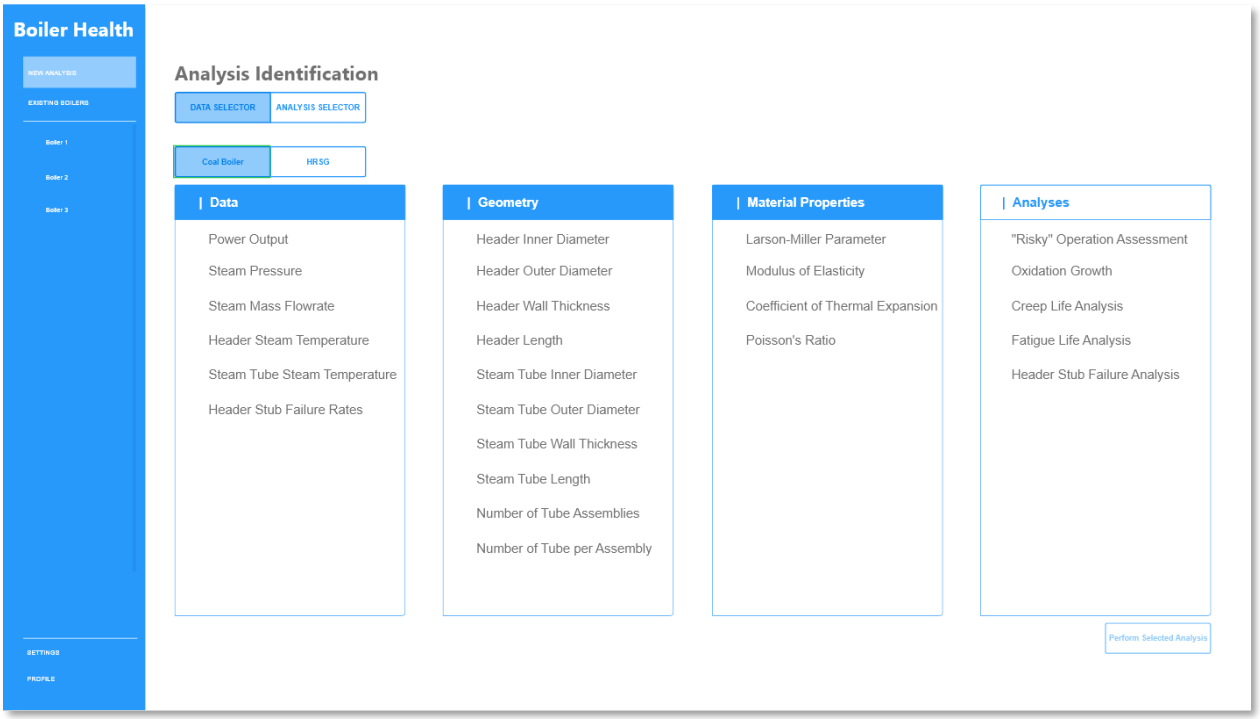


Fig. 6.55. An early mock-up of the graphical user interface which the user could use to select the desired analysis.

Implementation Roadmap

The roadmap for implementing the Superheater Damage Evaluation Software is structured around a one-year development period, with six months of initial development, followed by three months of testing, and an additional three months of development to incorporate user and quality assurance feedback. Development will be conducted in two-week sprints, with feature prioritization occurring at the start of each sprint based on a combination of technical dependencies and risk management consideration. A development timeline is shown below.

	Task Name	Hours	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
Analysis Development	Data Pre-processing	120												
	Cycle Counting Algorithm	80												
	"Risky" Operation Analysis	160												
	Oxide Growth Estimation	80												
	Creep Assessment	160												
	Fatigue Assessment	160												
	Header Stub Failure Rate Analysis	160												
GUI Development	GUI Framework	80												
	Database Structure	80												
	Data Upload and Database Insertion	80												
	Develop location mapping display	160												
	Analysis and Data Identification Interfaces	120												
	Developing reporting displays	120												
Preproduction Testing	SQA Submission & Testing	16												
	Member Testing	40												
Revision & Resubmission	Software Revision	160												
	Final SQA Submission	16												
	Total Estimated Hours	**1792**												

Fig. 6.56. A proposed timeline of software development activities and project duration for the tool and analyses described in **Appendix D.**

Key Milestones:

Months 1-6 (Development):

- Sprint planning every two weeks.
- Create and refine user stories and features; track and adjust development efforts as necessary.
- Core analysis and software interface can be developed in parallel.
- Proposed timeline:
  - Analysis Development:
    - Month 1: Begin implementing the required data cleaning checks and bad data handling processes.
    - Month 2: Implement the cycle counting algorithm. Finalize development of data cleaning and bad data handling processes.
    - Month 3: Begin implementing the “Risky” Operation analyses.
    - Month 4: Finalize development of the “Risky” Operation analyses. Integrate oxide growth estimation calculation. Begin implementing creep life analysis.
    - Month 5: Finalize development of the creep life analysis functionality. Begin implementing both the fatigue life and header stub failure rate analyses.
    - Month 6: Finalize development of core algorithms. Finalize development of the fatigue life and header stub failure rate analyses.
  - GUI Development:
    - Month 1: Develop graphical interface framework and begin building framework of databases.
    - Month 2: Finalize database structures and begin development of tools for data uploading and insertion into the appropriate databases.

    - Month 3: Finalize implementation of data upload and insertion and begin development of location mapping display and analysis and data identification tools.
    - Month 4: Finalize development of location mapping display and analysis and data identification tools.
    - Months 5-6: Begin developing appropriate displays for the different analyses as their respective analysis codes are completed.

Months 7-9 (Preproduction Testing by EPRI SQA and Users):

- Months 7-8 (SQA Testing)
    - Virus scanning.
    - Installation and compatibility testing.
    - Verification of software functionality against the user manual.
    - User manual tutorials validation.
- Months 8-9 (Preproduction Testing by Users):
    - Test users will interact with and utilize the software in their local environment.
    - Establish a straightforward feedback submissions process.

Months 10-12 (Revision and Final Production Submission):

- A single revision cycle based on SQA results and user feedback.
- Implement changes and prepare deliverables for final SWA submission, including:
    - Required SQA submission forms.
    - Updated software specification.
    - Software installation files.
    - Source code.
    - User manual with tutorials.

Post-Approval:

- Upon EPRI SQA approval, release the software for member use on EPRI's website.

This roadmap is designed to be simple, allowing for flexibility and adjustments as needed through the process. Some aspects of the software, namely the analysis code and the graphical interface, can be developed in parallel to ensure that overall progress is never blocked by any one issue. All relevant documentation and deliverables required for each phase of the SQA process have been specified, and a post-approval period has been included to guarantee a smooth transition into the release and onboarding of EPRI members.

7 Recommendations and research output

The hybrid physics-based and data-driven models described above broke ground with novel configurations of data processing to produce agile tools which accurately predict temperature distributions in the superheater of a coal fired boiler under real-world operation. These demonstrations show significant potential for hybrid approaches to further inform plant risk and reduce maintenance expenses. Deploying such robust models, however, requires significant up-front cost given that the highly instrumented plants leveraged in this project are the exception in the industry. While the project was extremely thankful to work with these plants, the appetite to implement complex life-prediction models on the average plant is significantly less, for reasons described above (Section 6.6.2).

Future work in this area needs to refine the hybrid model approach to maximize benefit with minimal work on behalf of the plant, both in terms of hardware installation (e.g., thermocouples) and engineering time to train and operate the model. With the 58 thermocouples installed on the superheater outlet header in Plant A, the model was able to accurately discretize the temperature distribution in the header to a fine granularity. Two questions remain regarding the value of incrementally more data sources: how is the capability of the model impacted by fewer thermocouples and what capability does the model pose when trained on Plant A, with all its thermocouples, to predict on Plant X which has significantly fewer? These discussions on cost-benefit could bridge a gap to help the hybrid model find a home.

Similarly, while the initial demonstrations have proven substantially accurate, the topic of validation has come up several times within the project's advisors. Given the nature of power plants and their operation, there is a healthy degree of conservative skepticism when new tools are introduced and the quickest way to kill an idea is for it to throw false positives. Validation in this arena is the process by which the model builds trust with its users. To that end, the recommendation is to continue to foster a relationship with demonstration opportunities and show correlation under various conditions between experimental results and model predictions. Regardless of the type of plant, tubing inspection today includes various metrics (particularly oxide thickness) which indicate to the maintenance team how to manage risk and component replacement. Those data compliment the temperature readings and associated model predictions in a way to inform risk and future action.

7.1 Research Output

7.1.1 Journal Publications

1. Mukherjee A, Bhattacharyya D, "Hybrid Series/Parallel All-Nonlinear Dynamic-Static Neural Networks: Development, Training, and Application to Chemical Processes", *Industrial and Engineering Chemistry Research* 2023, *62* (7), 3221–3237. https://doi.org/10.1021/acs.iecr.2c03339
2. Adeyemo S, Bhattacharyya D, "Optimal nonlinear dynamic sparse model selection and Bayesian parameter estimation for nonlinear systems", *Computers and Chemical Engineering* 2024, *180*, 108502. https://doi.org/10.1016/j.compchemeng.2023.108502
3. Saini V, Bhattacharyya D, Purdy D, Parker J, Boohaker C, "Nonlinear State Estimation of a Power Plant Superheater using the Extended Kalman Filter for Differential Algebraic Equation Systems", *Applied Thermal Engineering* 2024, *251*, 123471.

https://doi.org/10.1016/j.applthermaleng.2024.123471
4. Mukherjee A, Bhattacharyya D, "On the Development of Steady-State and Dynamic Mass-Constrained Neural Networks Using Noisy Transient Data", *Computers and Chemical Engineering* 2024, *187*, 108722. https://doi.org/10.1016/j.compchemeng.2024.108722
5. Mukherjee A*, Adeyemo S*, Bhattacharyya D, "All-Nonlinear Static-Dynamic Neural Networks versus Bayesian Machine Learning for Data-Driven Modelling of Chemical Processes", *The Canadian Journal of Chemical Engineering* 2024, In Press. https://doi.org/10.1002/cjce.25379.
6. Mukherjee A, Bhattacharyya D, "Development of Steady-State and Dynamic Mass-Energy Constrained Neural Networks using Noisy Transient Data", *CSChE Systems & Control Transactions* 2024, *3*, 330 – 337. https://doi.org/10.69997/sct.127425
7. Adeyemo S, Bhattacharyya D, "Development of Mass / Energy Constrained Sparse Bayesian Surrogate Models from Noisy Data", *CSChE Systems & Control Transactions* 2024, *3*, 99 – 104. https://doi.org/10.69997/sct.10194
8. Mukherjee A, Bhattacharyya D, "Development of Steady-State and Dynamic Mass and Energy Constrained Neural Networks for Distributed Chemical Systems Using Noisy Transient Data", *Industrial and Engineering Chemistry Research* 2024, *63* (32), 14211–14239. https://doi.org/10.1021/acs.iecr.4c01429
9. Mukherjee A*, Saini V*, Adeyemo S*, Bhattacharyya D, Purdy D, Parker J, Boohaker C, "Development of Hybrid First Principles – Artificial Intelligence Dynamic Models for Applications to Power Plant Monitoring under Load-Following Operation", *Applied Thermal Engineering*, Under Review.
10. Mukherjee A, Gupta D, Bhattacharyya D, "Constrained Hybrid Gaussian Radial Basis Neural Networks: Development, Training, and Applications to Modeling Nonlinear Dynamic Processes", *IEEE Transactions on Neural Networks and Learning Systems*, Under Review.
11. Adeyemo S, Bhattacharyya D, "Sparse Mass-Constrained Nonlinear Dynamic Model Building from Noisy Data Using a Bayesian Approach", *Industrial and Engineering Chemistry Research*, Under Review.
12. Mukherjee A, Bhattacharyya D, "Development of Mass, Energy, and Thermodynamics Constrained Steady-State and Dynamic Neural Networks for Interconnected Chemical Systems", *Chemical Engineering Science*, Under Review.
13. Adeyemo S, Bhattacharyya D, "Sparse Modeling of Nonlinear, Dynamic, Time-Delay Systems Using Bayesian Brain Hypothesis", to be submitted to *AICHE Journal*.
14. Saini V, Mukherjee A, Adeyemo S, Bhattacharyya D, Purdy D, Lolla T, Parker J, Boohaker C, "Development of an Online Health Monitoring Framework for High Temperature Boiler Components by using Hybrid First Principles – Machine Learning Models", *Manuscript under preparation*.
15. Mukherjee A, Bhattacharyya D, "Optimal Discovery of Hybrid First-Principles Data-Driven Models for Process Systems Applications using Superstructure Optimization", *Manuscript under preparation*.
16. Mukherjee A, Bhattacharyya D, "Data-Driven Modeling of Complex Transient Chemical Processes using Hybrid Physics-Constrained Deterministic-Stochastic Neural Networks", *Manuscript under preparation*.
17. Adeyemo S, Bhattacharyya D, "Bayesian Development of Sparse Mass and Energy-Constrained Nonlinear Dynamic Models from Noisy Data", *Manuscript under preparation*.

18. Saini V, Purdy D, Lolla T ,Bhattacharyya D, Parker J, Boohaker C, “Prediction of oxide scale and tube life consumption for a superheater system using hybrid grey box modeling”, *Manuscript under preparation*.
19. Saini V, Bhattacharyya D,, “Development of mass and energy constrained estimation algorithms for dynamic distributed process systems”, *Manuscript under preparation*.

* These authors shared equal contributions.

7.1.2 Conference Presentations

1. Saini V, Mukherjee A, Adeyemo S, Bhattacharyya D, “Development of an Online Health-Monitoring Framework for High-Temperature Boiler Components by using Hybrid First Principles – Data Driven Models”, Paper 314f, AIChE Annual Meeting, San Diego, CA, October 27-31, 2024.
2. Mukherjee A, Bhattacharyya D, “Development of Steady-State and Dynamic Mass-Energy-Thermodynamics Constrained Neural Network (MET-CNN) Models for Interconnected Systems using Noisy Transient Data”, Paper 14i, AIChE Annual Meeting, San Diego, CA, October 27-31, 2024.
3. Adeyemo S, Bhattacharyya D, “Optimal Selection of Nonlinear Dynamic Sparse Models with Exact Satisfaction of Mass and Energy Constraints”, Paper 375ah, AIChE Annual Meeting, San Diego, CA, October 27-31, 2024.
4. Mukherjee A, Bhattacharyya D, “Development of Steady-State and Dynamic Mass-Energy Constrained Neural Network Models using Noisy Temporal Data for Dynamic Optimization of Distributed Chemical Systems”, Paper 676g, AIChE Annual Meeting, San Diego, CA, October 27-31, 2024.
5. Saini V, Bhattacharyya D, “Development of Constrained Estimation Algorithms for Distributed Parameter Systems for Satisfying Mass and Energy Conservation”, Paper 518a, AIChE Annual Meeting, San Diego, CA, October 27-31, 2024.
6. Mukherjee A, Bhattacharyya D, “Hybrid Gaussian Radial Basis Neural Networks (GRAB-NN): Development, Training, and Applications to Modeling Nonlinear Dynamic Noisy Chemical Processes”, Paper 674d, AIChE Annual Meeting, San Diego, CA, October 27-31, 2024.
7. Saini V, “Dynamic Modeling and Estimation for Condition Monitoring of Energy Systems”, Paper 364s, AIChE Annual Meeting, San Diego, CA, October 27-31, 2024.
8. Adeyemo S, “New Approaches to Surrogate Modeling under Uncertainty and Adaptive Learning in Systems Engineering”, Paper 4ly, AIChE Annual Meeting, San Diego, CA, October 27-31, 2024.
9. Mukherjee A, Bhattacharyya D, “Development of Steady-State and Dynamic Mass-Energy Constrained Neural Networks using Noisy Transient Data”, Foundations of Computer-Aided Process Design (FOCAPD) Conference, Breckenridge, CO, July 14-18, 2024.
10. Adeyemo S, Bhattacharyya D, “Development of Mass / Energy Constrained Sparse Bayesian Surrogate Models from Noisy Data”, Foundations of Computer-Aided Process Design (FOCAPD) Conference, Breckenridge, CO, July 14-18, 2024.
11. Mukherjee A, Saini V, Adeyemo S, Bhattacharyya D, “Development of Hybrid First Principles – Artificial Intelligence Models: Application to an Industrial Steam Superheater System”, Paper 374d, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.

12. Saini V, Mukherjee A, Adeyemo S, Bhattacharyya D, “Grey-Box Modeling and Estimation for Health Monitoring of High Temperature Boiler Components”, Paper 433a, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.
13. Adeyemo S, Bhattacharyya D, “Identification of Mass-Constrained Bayesian Sparse Models from Noisy Data”, Paper 199e, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.
14. Saini V, Bhattacharyya D, “Development of Mass and Energy Constrained Estimator Algorithms for Process Systems”, Paper 295f, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.
15. Mukherjee A, “New Data-Driven Modeling Paradigms in Systems Engineering using Novel Neural Network Structures”, Paper 2dw, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.
16. Saini V, "Hybrid Grey-Box Modeling and Estimation for Condition Monitoring of Complex Dynamics Energy Systems", Paper 147n, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.
17. Adeyemo S, "Development of Algorithms for Identification of Sparse Dynamic Models with Mass and Energy Conservation from Noisy Data", Paper 147g, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.
18. Mukherjee A, Bhattacharyya D, “Development of Algorithms for Mass and Energy Constrained Dynamic Neural Network Models”, Paper 59o, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.
19. Adeyemo S, Bhattacharyya D, “Bayesian Identification of Nonlinear, Sparse, Dynamic Models”, Paper 149ae, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.
20. Mukherjee A, Bhattacharyya D, “Hybrid Series/Parallel All-Nonlinear Dynamic-Static Stochastic Neural Networks: Development, Training and Application to Chemical Processes”, Paper 149x, AIChE Annual Meeting, Orlando, FL, November 5-10, 2023.
21. Mukherjee A, Bhattacharyya D, “Hybrid Series and Parallel All-Nonlinear Dynamic-Static Neural Networks: Development, Training, and Application to Chemical Processes”, AIChE Midwest Regional Conference, Chicago, IL, April 11-12, 2023.
22. Mukherjee A, Bhattacharyya D, “Development of Algorithms for Mass-Constrained Dynamic Neural Networks”, Foundations of Process/Product Analytics and Machine Learning (FOPAM) Conference, UC Davis, CA, July 30 - Aug 3, 2023.
23. Bhattacharyya D, Mukherjee A, Adeyemo S, Saini V, Parker J, Boohaker C, Purdy D, “Boiler Health Monitoring Using a Hybrid First Principles-Artificial Intelligence Model”, EPRI Generation & Reliability Technical Week, Digital Demonstration Facility Workshop, Orlando, FL, July 10-14, 2023.
24. Mukherjee A, Bhattacharyya D, “Development of Mass and Energy Constrained Neural Networks”, Paper 12b, AIChE Annual Meeting, Phoenix, AZ, November 13-18, 2022.
25. Saini V, Bhattacharyya D, “State and Parameter Estimation of Complex Dynamic Systems Using Nonlinear Grey Box Modeling: Application to an Industrial Steam Superheater System”, Paper 484g, AIChE Annual Meeting, Phoenix, AZ, November 13-18, 2022.
26. Mukherjee A, Bhattacharyya D, “Data-Driven Modeling of Complex Nonlinear Systems using Hybrid Series and Parallel Nonlinear Static - Nonlinear Dynamic Stochastic Neural Networks”, Paper 362p, AIChE Annual Meeting, Phoenix, AZ, November 13-18, 2022.
27. Adeyemo S, Bhattacharyya D, "Development of Bayesian Machine Learning Algorithms for Optimal Nonlinear Model Selection and Parameter Estimation from Noisy Data of Unknown Characteristics", Paper 362f, AIChE Annual Meeting, Phoenix, AZ, November 13-18, 2022.

28. Mukherjee A, Bhattacharyya D, “Data-Driven Modeling of Complex Nonlinear Systems using Hybrid Series and Parallel Nonlinear Static – Nonlinear Dynamic Neural Networks”, AIChE Advanced Manufacturing and Processing Conference, Bethesda, MD, June 1-3, 2022.
29. Mukherjee A, Bhattacharyya D, “Modeling Complex Nonlinear Systems using Concatenated Static-Dynamic Neural Networks”, Paper 415a, AIChE Annual Meeting, Boston, MA, November 7-19, 2021.
30. Saini V, Reynolds K, Bhattacharyya D, “Condition Monitoring of Non-Linear Differential-Algebraic Equation Systems with Colored Process and Measurement Noise”, Paper 346t, AIChE Annual Meeting, Boston, MA, November 7-19, 2021.
31. Adeyemo S, Bhattacharyya D, “Bayesian Machine Learning of Dynamic Systems with Unknown Noise Characteristics”, Paper 346o, AIChE Annual Meeting, Boston, MA, November 7-19, 2021.
32. Bhattacharyya D, “Boiler Health Modeling Using a Hybrid First-Principles Artificial Intelligence Model”, 2021 Sensors and Control Review Meeting, Virtual, May 18-19, 2021.
33. Mukherjee A, Bhattacharyya D, “Development of a Hybrid First Principles – Artificial Intelligence Approach for Dynamic Modeling of Complex Systems”, Virtual AIChE Annual Meeting, November 16-20, 2020.
34. Saini V, Bhattacharyya D, “Nonlinear State and Parameter Estimation for Differential-Algebraic Equation Systems with Uncertain Algebraic Equations and Unknown Noise Covariance”, Virtual AIChE Annual Meeting, November 16-20, 2020.
35. Bhattacharyya D, “Boiler Health Monitoring Using a Hybrid First Principles-Artificial Intelligence Model”, 2020 Sensors and Controls Project Review Meeting, Virtual, August 26, 2020.

7.1.3 Public GitHub Repositories

1. The MATLAB codes developed for training and validation of the proposed **hybrid all-nonlinear series and parallel static-dynamic NN models** can be publicly accessed on GitHub at:https://github.com/mukherjeeangan26/Hybrid_StaticDynamic_NN_Models.
2. The MATLAB codes developed for training and validation of the proposed **Bayesian Identification of Dynamic Sparse Algebraic Models (BIDSAM)** can be publicly accessed on GitHub at: https://github.com/adeyesam/BML_Codes.
3. The Python and MATLAB codes for **steady-state and dynamic MCNNs** developed in this work can be found on GitHub at: https://github.com/mukherjeeangan26/Steady_Dynamic_MCNN.
4. The Python and MATLAB codes for **steady-state and dynamic ECNNs** (MECNNs) developed in this work can be found on GitHub at: https://github.com/mukherjeeangan26/Steady_Dynamic_ECNN.
5. The Python / Pyomo notebooks for development of different **RBF / GRAB-NN / McRBF model structures and training algorithms** can be accessed on GitHub via the following link: https://github.com/mukherjeeangan26/Hybrid_RBF_GRABNN.

Appendices

Appendix A: Additional validation results for the first principles model and nonlinear estimator for Plant B superheater system

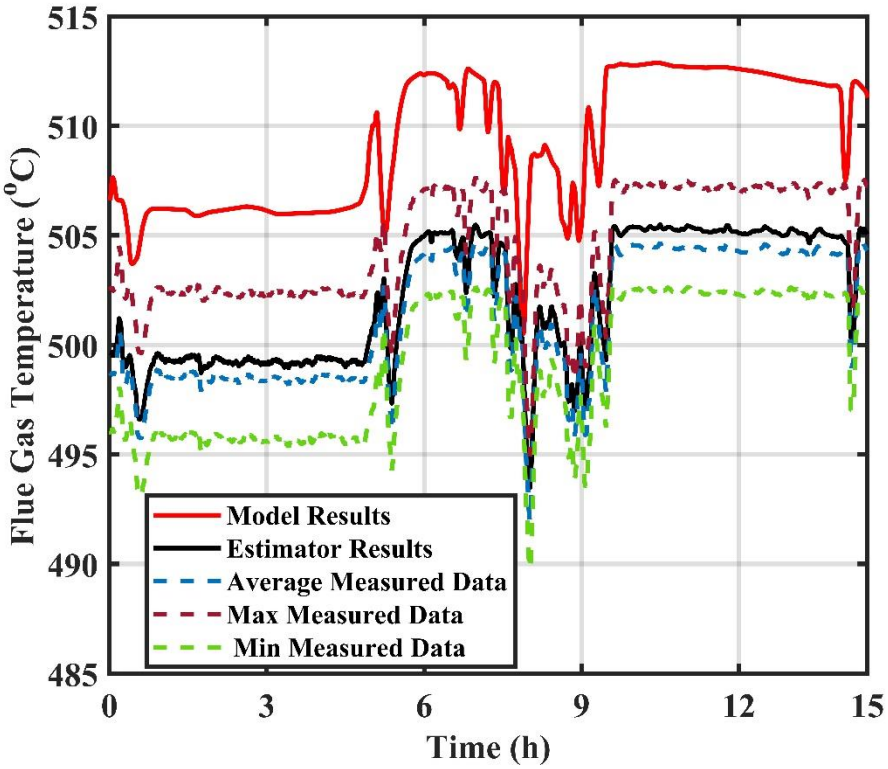


Fig. A. 1. Comparison of the model and estimator results with the measured data for the flue gas temperature at an elevation of 71 feet (G 71').

Fig. A. 1 compare the measurement data of flue gas temperature at location G 71' (see Fig. 2.2**)** with the model and estimator results. Fig. A. 1 shows large mismatch of model results from the average measured as well as maximum and minimum measured data for the entire duration of the dataset considered. The model is overpredicting all the available measured data at this particular location. However, the estimator results are found to be close to the average measured values even in the presence of large model mismatch.

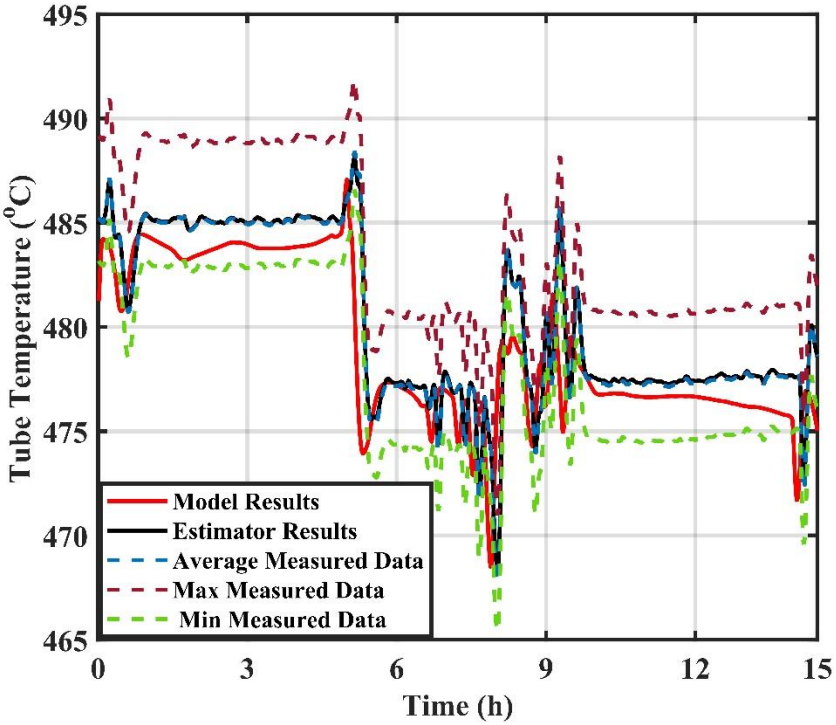


Fig.A. 2. Comparison of the model and estimator results with the measured data for the tube temperature at an elevation of 49 feet (E 49').

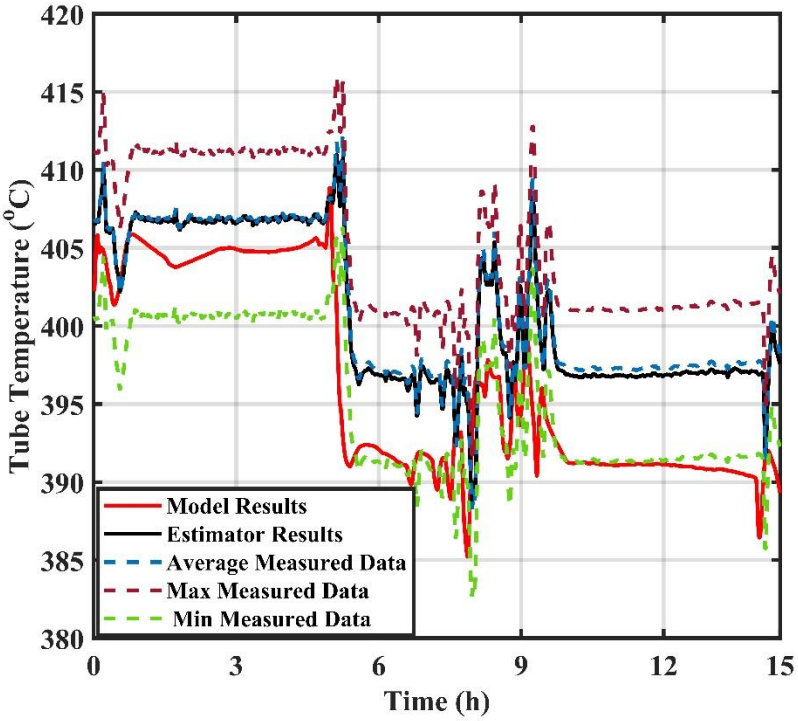


Fig.A. 3. Comparison of the model and estimator results with the measured data for the tube temperature at return bend 2 (RB 2).

Fig.A. 2 and Fig.A. 3 compare the measurement data of tube wall temperature at location E 49’ and RB 2 (see Fig. 2.2**),** respectively**,** with the model and estimator results. In both figures, the model deviates from the average measured data but is within the bounds of maximum and minimum measured data. However, in Fig.A. 3 the model is hitting the minimum bound between 9-15 h**.** The application of the estimator is very useful in predicting the actual average measured values for both variables in the entire operating time even in the presence of the significant model mismatch.

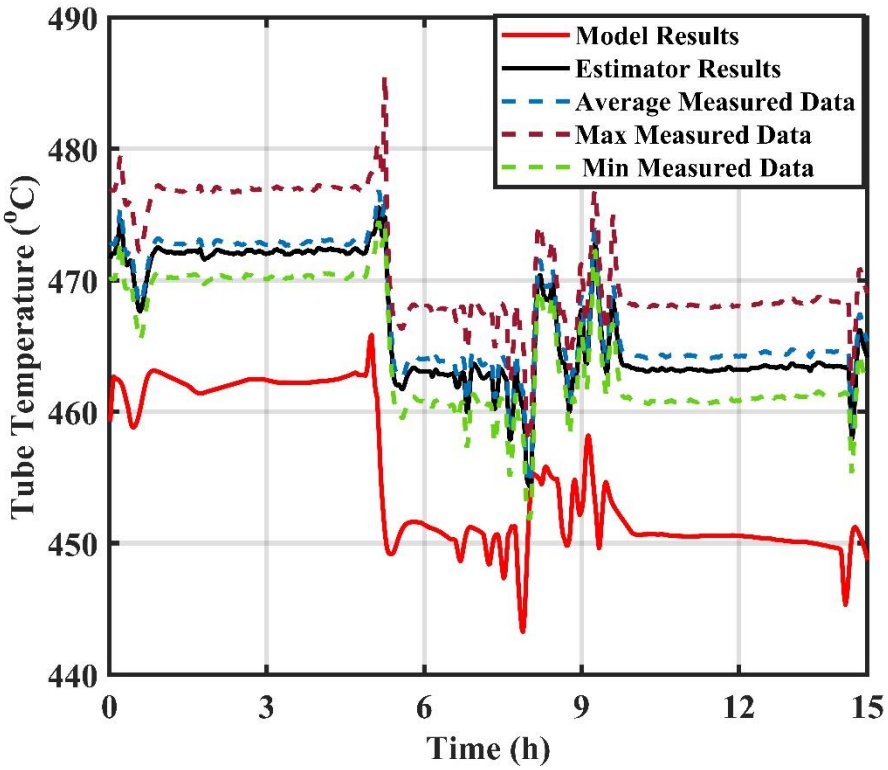


Fig.A. 4. Comparison of the model and estimator results with the measured data for the tube temperature at return bend 1 (RB 1).

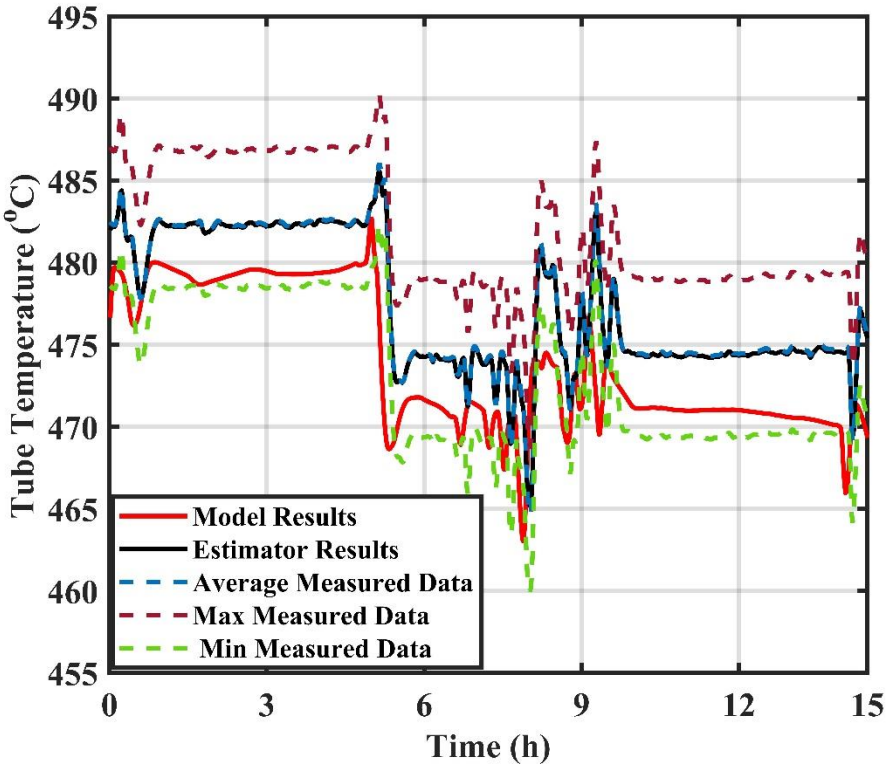


Fig.A. 5. Comparison of the model and estimator results with the measured data for the tube temperature an elevation of 6 feet (E 6').

Fig.A. 4 and Fig.A. 5 compare the measurement data of tube wall temperature at location RB 1 and E 6' (see Fig. 2.2**),** respectively**,** with the model and estimator results. Again, the model has large deviation from the actual measured data for both these locations. In Fig.A. 4**,** the model underpredicts the available measured data while in Fig.A. 5**,** th**e** minimum bound is hit by the model**.** These deviations are reduced by using the estimator, which corrects the model-measured data mismatch and provides the optimum temperature profile near the observed value.

The root mean square error (RMSE) of the model and estimator results with respect to the measurement data for all five variables is reported in **Table A.1** which shows that the model error was reduced by about 80% for all five variables when the estimator was used.

Table A.1: RMSE for five measured variables for the final superheater system (NGCC plant)

Root Mean Square Error (RMSE)		
Variable name	**Model RMSE error (ºC) (Measured-Model)**	**Estimator RMSE error (ºC) (Measured-Estimated)**
Flue Gas Temperature (G 71')	8.19	0.83
Tube Temperature (E 49')	2.4	0.14
Tube Temperature (RB 2)	5.68	0.41
Tube Temperature (RB 1)	12.92	0.93
Tube Temperature (E 6')	4.15	0.17

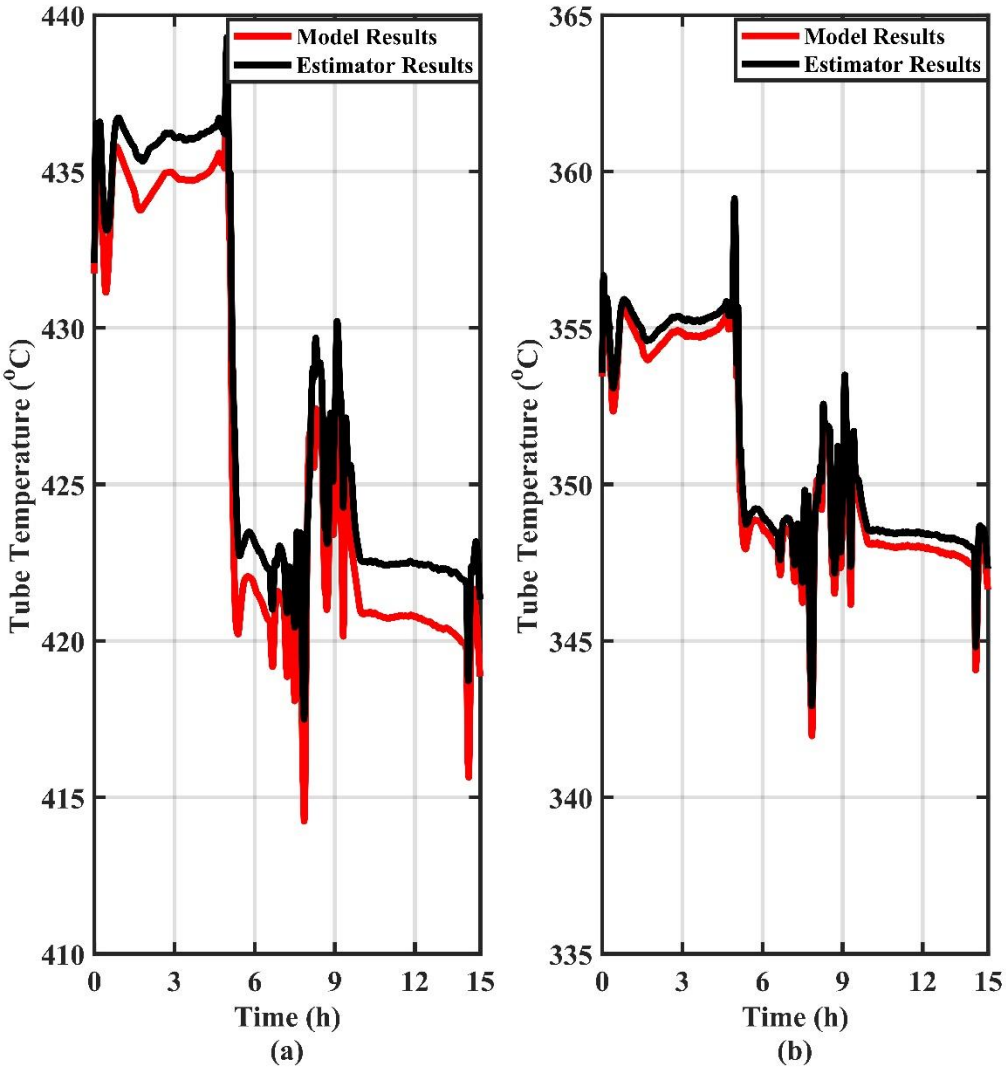


Fig.A. 6. Estimation of unmeasured intermediate headers outer tube wall temperatures (a) Intermediate Header 2 outer tube wall temperature (b) Intermediate Header 3 outer tube wall temperature (see Fig. 2.2**).**

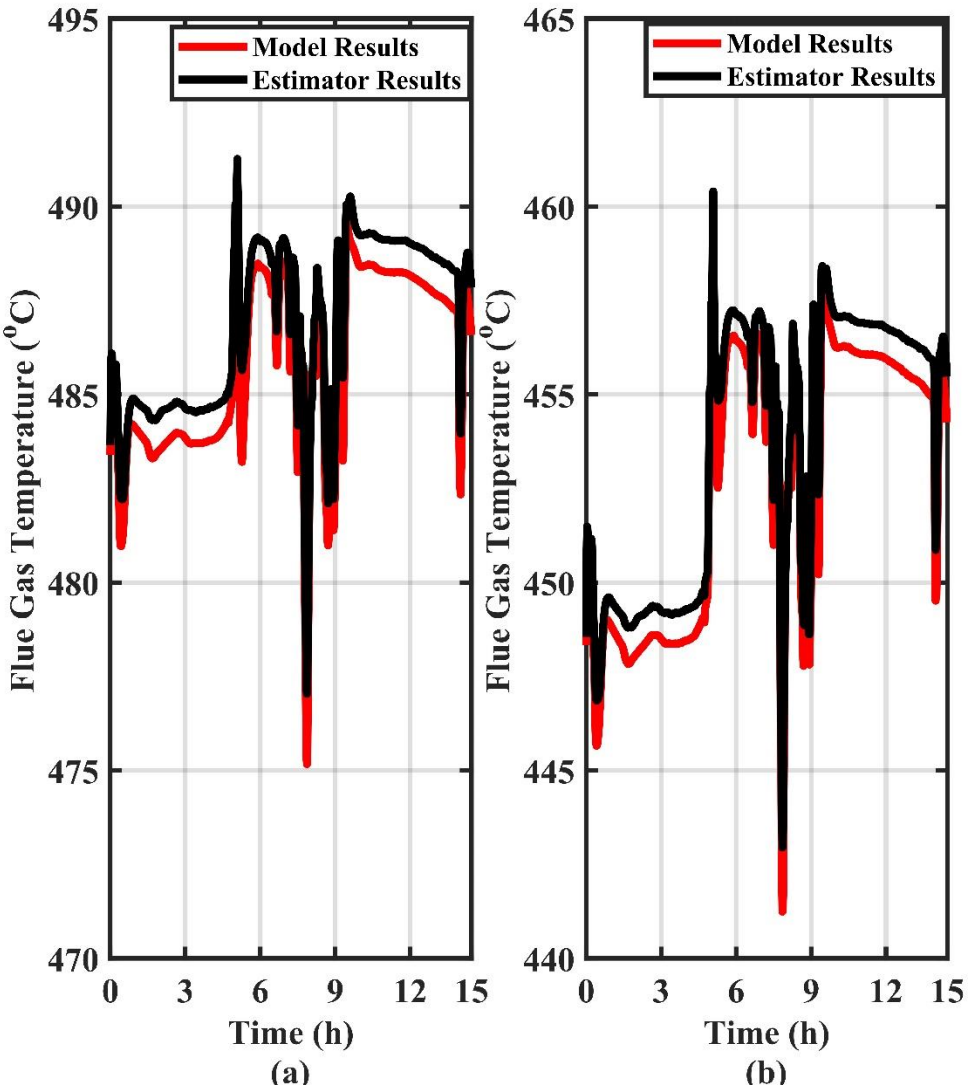


Fig.A. 7. Estimation of unmeasured intermediate headers flue gas temperatures.
(a) Intermediate Header 2 flue gas temperature (b) Intermediate Header 3 flue gas temperature (see Fig. 2.2**).**

The utility of using a first principles model and estimator lies in its abiltiy to estimate unmeasured variables accurately**.** Fig.A. 6 and Fig.A. 7 compares the model and estimator results of the tube wall and flue gas temperatures at the intermediate headers location for the superheater system **(**see Fig. 2.2**).** These variables are difficult to measure by placing a sensor due to the curved geometry of the headers and accessibility issues in tightly placed tube bundles.

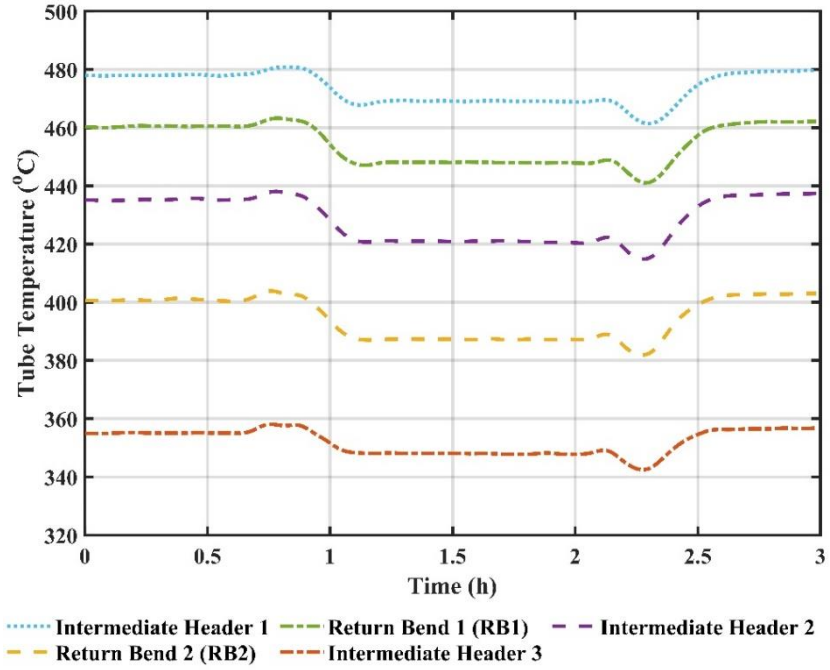


Fig.A. 8. Estimated outer tube wall temperature profiles along the intermediate headers and return bends (see Fig. 2.2**).**

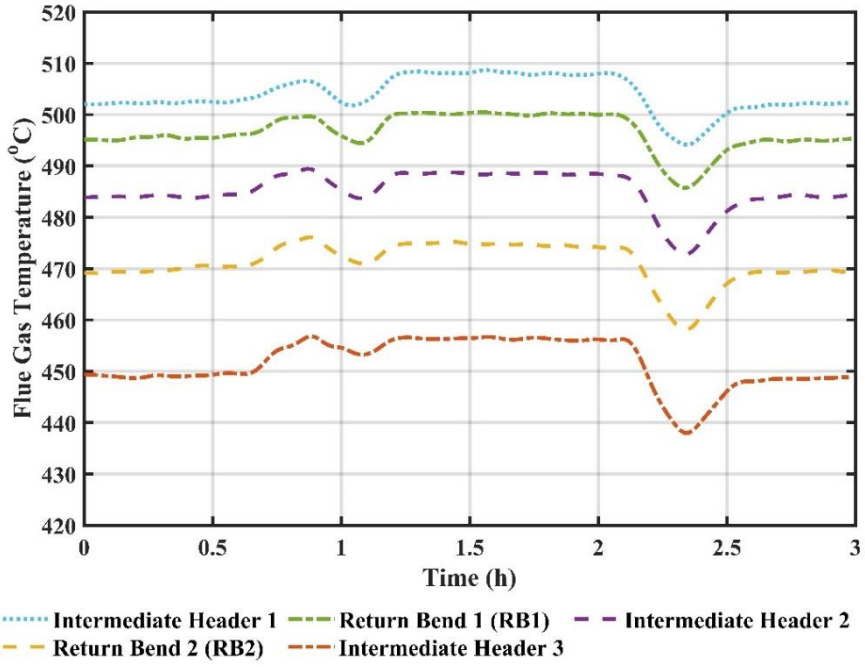


Fig.A. 9 . Estimated flue gas temperature profile along the intermediate headers and return bends (see Fig. 2.2**).**

Other unmeasured variables of interest whose results are provided by estimator are shown in Fig.A. 8 and Fig.A. 9**.** Here, the estimated values of temperature distribution in intermediate headers and return bends for outer tube walls and flue gas temperatures are shown. The accurate prediction of these variables is very critical as the headers and bends are often the places with large thermal stresses. However, obtaining sensor data at these places is difficult due to accessibility issues. From both the Fig.A. 8 and Fig.A. 9**,** we observe that Intermediate Header 3 has the lowest tube wall and flue gas temperatures and Intermediate Header 1 has the highest. It is observed that the magnitude of temperature increase between Intermediate Header 3 and Return Bend 2 is the highest due to the temperature difference driving force being the highest there as the steam enters at Pass 6. On the other hand, the driving force is the lowest towards the Intermediate Header 1 (near Pass 1).

Appendix B : Additional results for the development and validation of the Bayesian ML Model

Modeling of nonisothermal CSTR

Table B. 1. Selected basis functions for BIDSAM and ALAMO steady state models for non-isothermal Van de Vusse reactor.

Model	No. of basis functions	Selected basis Functions
True data		
BIDSAM	4	u_2, $\sqrt{u_1}$, const., $\sqrt{u_3}$
ALAMO A	15	u_1, u_2, u_3, $logu_3$, e^{u_1}, e^{u_2}, u_1^2, e^{u_3}, u_2^2, u_1u_2, u_1u_3, u_2u_3, $logu_1$, u_3^2, 1
ALAMO B	18	u_2, u_1u_2, u_1u_3, u_2u_3, u_2^2, u_3^2, $\frac{1}{u_2}$, $\frac{1}{u_1^2}$, $\sqrt{u_1}$, $\frac{1}{\sqrt{u_3}}$, u_1, $\sqrt{u_3}$, $\frac{1}{\sqrt{u_1}}$, u_3, $\frac{1}{u_1}$, $\frac{1}{u_3}$, 1, $logu_3$
Noisy data		
BIDSAM	4	u_2, $\sqrt{u_1}$, const., $\sqrt{u_3}$
ALAMO A	10	u_2, $logu_3$, u_1^2, u_3^2, u_1u_2, u_2u_3, u_1, e^{u_3}, 1, u_3,
ALAMO B	16	u_2, u_1u_2, u_2u_3, u_1^2, $logu_1$, $\frac{1}{u_3^2}$, u_1u_3, $\frac{1}{\sqrt{u_1}}$, $\frac{1}{1+e^{-u_1}}$, 1, $logu_3$, $\frac{1}{u_3}$, u_1, u_3^2, e^{-u_3}, $\frac{1}{u_3}$

Table B. 2. Training time analyses for developing steady state model for van de Vusse reactor.

Step	**Sub-steps**	**Time taken**
Data preparation	• Normalization • Population of library of basis functions + tracking of indices for interpretation • Selection of training/validation data	~0.0135 sec
Ranking of basis functions	• MAP by EM algorithm for 30 models with individual basis functions	~0.6395 sec
Single model		
Candidate model estimation	• MAP by EM algorithm	~0.1095 sec
Computation of objective function		~0.0053 sec
Branch and bound	• Parallelization • MAP by EM algorithm • Bookkeeping	**Parallel computation (8 workers)** • Total time=1080 sec (18min) • Nodes explored ~ (not tracked) • All possible nodes = 1.0737e9 • % explored = NA
		Nonparallel (single worker) • Total time=8580 sec (143min) • Nodes explored = 66617 • All possible nodes = 1.0737e9 • % explored = 6.20e-3% • Time/node ~ 0.1288sec

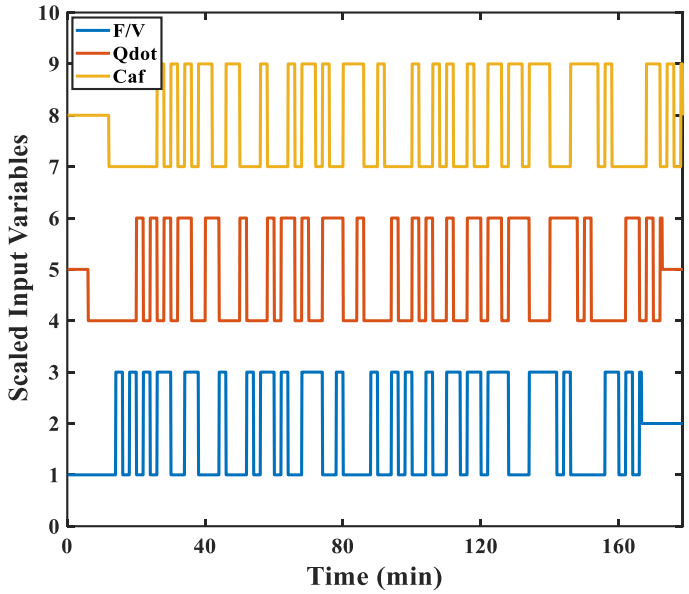


Fig. B. 1. PRBS signal for generating dynamic data for Van de Vusse reactor.

Table B. 3. Selected basis functions for some of the estimated models for non-isothermal Van de Vusse reactor.

Model	No. of basis functions	Selected basis Functions
		True data
BIDSAM I	13	$y_1, y_2, y_3, y_4, \frac{1}{1+e^{-u_3}}, u_2u_3, u_3, \sqrt{u_2}, 1, u_2^2, u_1u_3, logu_1, u_1y_4$
BIDSAM II	11	$y_1, y_2, y_3, y_4, \sqrt{u_3}, u_2u_3, \sqrt{u_2}, u_3^2, logu_2, \frac{1}{1+e^{-u_1}}, u_1u_3$
SINDy A2	21	$1, y_1, y_2, y_4, u_1, u_3, y_1^2, y_1y_3, y_1y_4, y_2y_4, y_2u_3, y_3, u_1, y_3^2, y_1y_2, y_2^2, y_2y_3, y_3y_4, y_4^2, u_2, u_1^2$
SINDy B	48	$e^{-y_1}, y_1, y_2, u_1, u_3, y_1^2, y_1y_3, y_1y_4, y_2y_4, logy_1, logy_2, \sqrt{y_1}, \sqrt{y_2}, \sqrt{y_3}, \frac{1}{\sqrt{y_1}}, \frac{1}{\sqrt{y_4}}, \frac{1}{1+e^{-y_1}}, e^{-y_2}, \frac{1}{y_2}, \frac{1}{y_4}, \frac{1}{u_1}, y_4, y_3, y_2^2, 1, logy_2, logy_4, \frac{1}{y_2^2}, \frac{1}{u_1^2}, \sqrt{u_1}, \frac{1}{\sqrt{y_2}}, \frac{1}{\sqrt{y_4}}, \frac{1}{\sqrt{u_1}}, \frac{1}{1+e^{-y_2}}, \frac{1}{1+e^{-y_3}}, e^{-y_2}, e^{-y_4}, u_2, y_4^2, y_2y_3, logu_2, logu_3, logu_4, \frac{1}{y_4^2}, \frac{1}{u_1^2}, \sqrt{y_4}, \frac{1}{\sqrt{y_2}}, \frac{1}{1+e^{-y_4}},$
		Noisy data
BIDSAM I	10	$y_1, y_2, y_3, y_4, \sqrt{u_3}, u_2u_3, u_1u_3, \frac{1}{\sqrt{u_2}}, \frac{1}{u_1^2}, e^{-u_1}$
BIDSAM II	10	$y_1, y_2, y_3, y_4, \sqrt{u_3}, u_2u_3, u_1u_3, \frac{1}{\sqrt{u_2}}, e^{-u_1}, \frac{1}{u_1}$

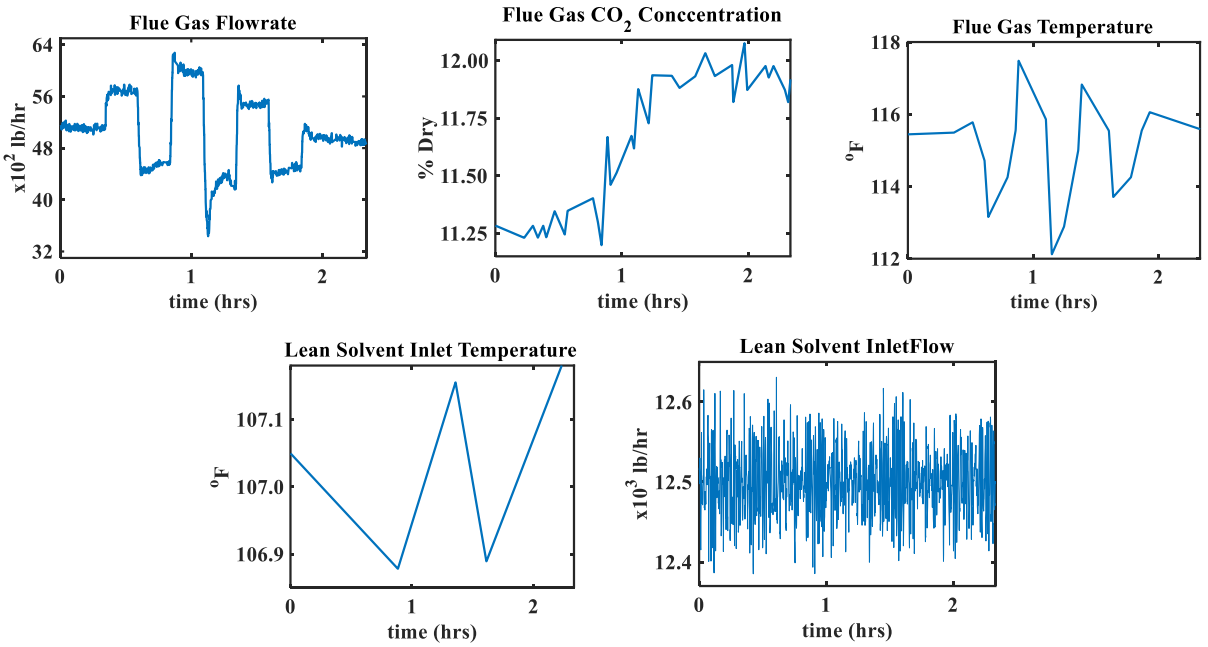


Fig. B. 2. Transients in all key input variables during step tests in the flue gas flowrate for CO2 absorber.

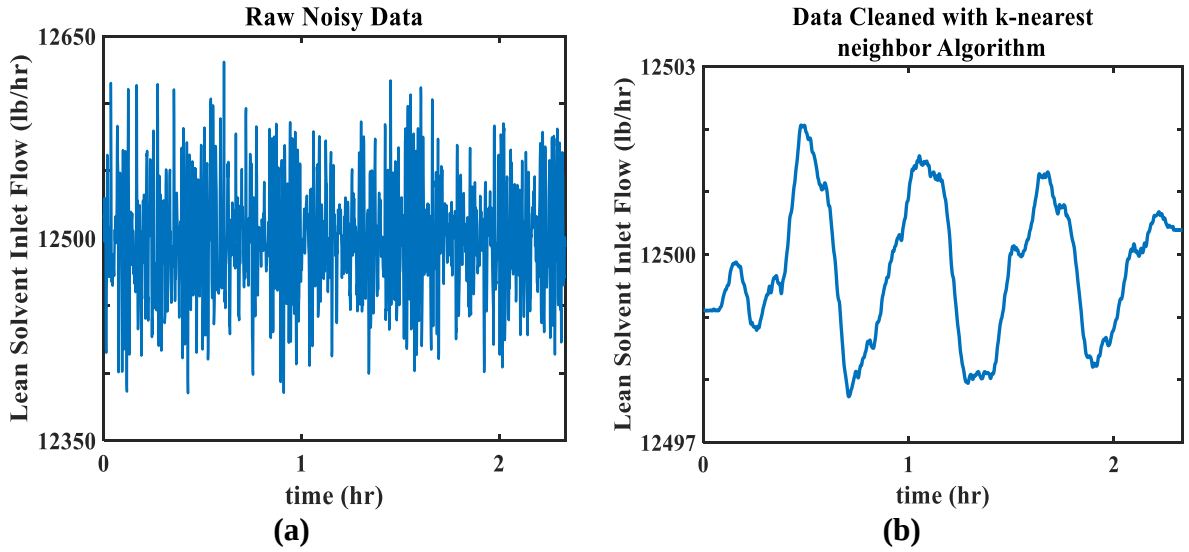


Fig. B. 3. Lean solvent inlet flow rate as input variable (a) raw plant measurement (b) denoised data using k-nearest neighbor algorithm.

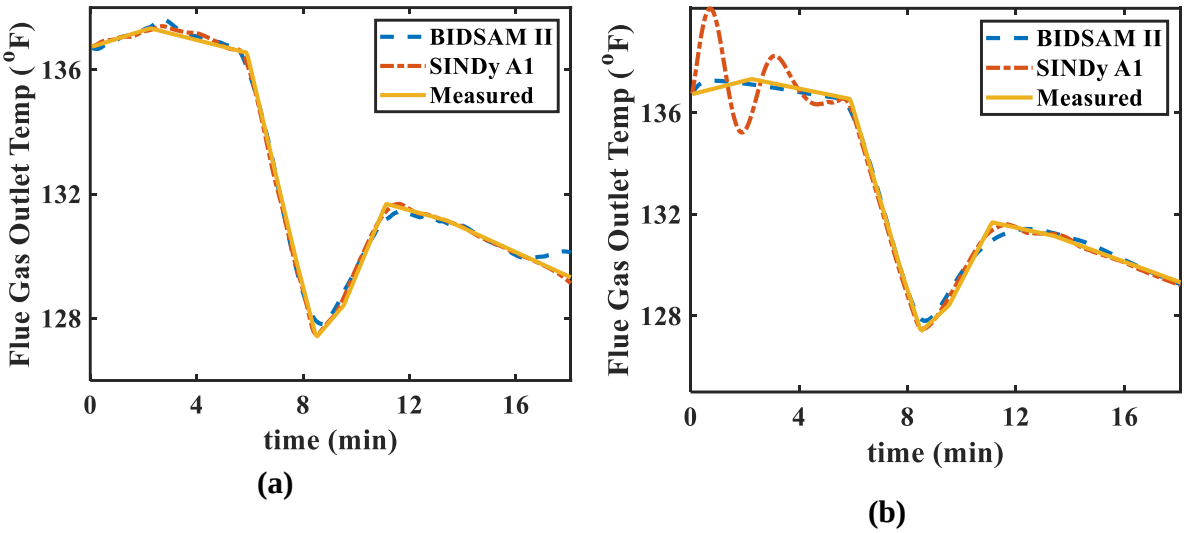


Fig. B. 4. Simulation plots for models trained from single step change in flue gas flow rate (a) noise removed from data (b) noise not removed from data.

Table B. 4. Performance indices of identified models for the step changes in the flue gas flow rate for CO2 absorber.

Model	No. of Parameters	% RMSE		AIC_c
		T_{fg}	CO_2 Conc.	
Single step change in flue gas flow, ***noise removed from data***				
BIDSAM II	16	0.17	3.56	-477.5
SINDy A1	34	0.09	1.51	-448.0
Single step change in flue gas flow, ***without removing noise from data***				
BIDSAM II	18	0.16	3.25	-474.2
SINDy A1	33	0.55	3.01	-356.2
Multiple step changes in flue gas flow, ***noise removed from data***				
BIDSAM II	40	0.34	7.72	-3552
SINDy A1(multiple time prediction)	60	-	-	-
SINDy A1(single time prediction)	60	0.03	0.36	-3974

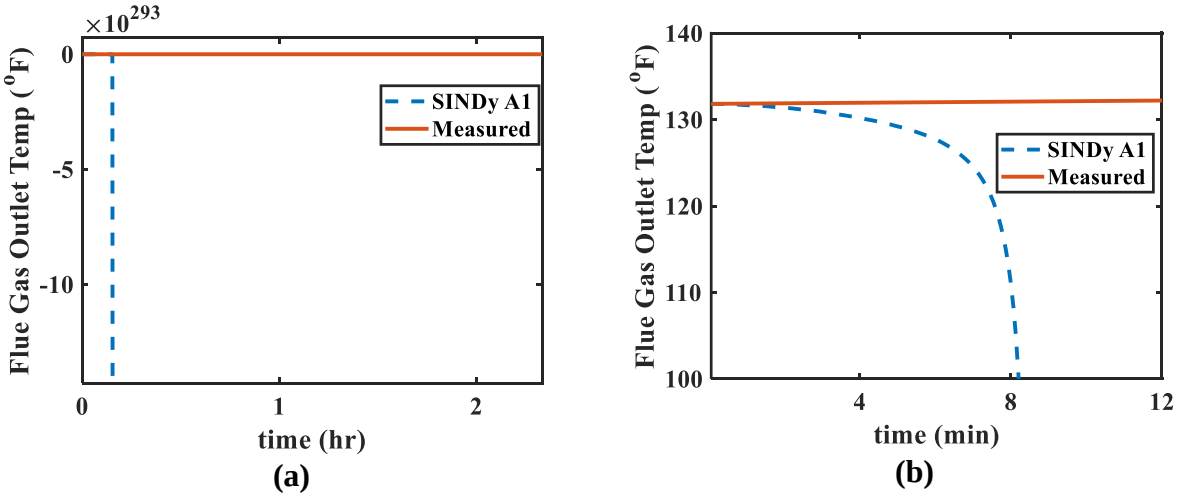


Fig. B. 5. Simulation plots for models trained from data for multiple step changes in flue gas flow (a) SINDy A1 model (b) zoomed portion of SINDy A1 model validation.

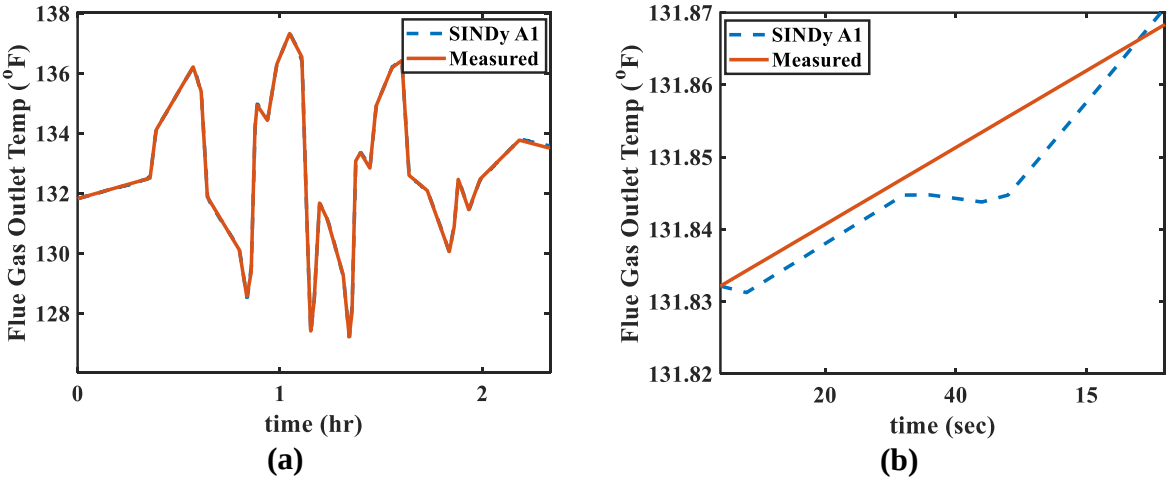


Fig. B. 6. Simulation plot for SINDy A1 model with all successive predictions based on measurements (b) zoomed portion of the validation plot.

Table B. 5. Selected basis functions for BIDSAM models for multiple step changes in flue gas flow rate.

Model	No. of basis functions	Selected basis Functions
Measured data		
BIDSAM I	12	$\frac{1}{1+e^{-u_1}}, \frac{1}{u_5}, 1, \frac{1}{1+e^{-u_2}}, u_1u_5, \sqrt{u_2}, u_1u_2, \frac{1}{1+e^{-u_5}}, u_5y_2, u_5y_1, logu_5, \frac{1}{u_3}$
BIDSAM II	18	$u_1u_3, \frac{1}{1+e^{-u_1}}, \sqrt{u_3}, \frac{1}{1+e^{-u_3}}, u_3, \frac{1}{u_5}, \frac{1}{1+e^{-u_2}}, \frac{1}{1+e^{-u_4}}, \sqrt{u_2}, \frac{1}{u_2}, u_5, u_5y_2, u_5y_1, u_2u_5, logu_2, u_2y_1, logu_5, u_3y_1$
Measured data + Noise		
BIDSAM I	19	$u_1u_3, \sqrt{u_5}, \sqrt{u_3}, \frac{1}{u_1}, u_2u_5, u_5^2, u_4^2, u_1y_2, u_2u_4, e^{-u_2}, \frac{1}{u_2^2}, u_2^2, u_1y_1, e^{-u_1}, \frac{1}{u_3}, u_2y_2, u_2y_1, u_4y_1$
BIDSAM II	20	$\sqrt{u_3}, u_3, \frac{1}{1+e^{-u_5}}, u_1u_5, \frac{1}{\sqrt{u_4}}, \frac{1}{\sqrt{u_2}}, \frac{1}{\sqrt{u_1}}, \frac{1}{u_1}, u_4u_5, u_2u_3, u_2, u_5^2, u_4^2, logu_4, u_2u_4, logu_2, u_2y_1, u_4y_2, u_4y_1, \frac{1}{u_1^2},$

Derivation of constraints for Algorithm II

PART 1: Without Hold-up Information

For dynamic modeling of the system in Fig. 3.20, in place of model given by Eq.(3.46) we have:

$$\begin{pmatrix} y_{1,k+1} \\ y_{2,k+1} \\ y_{3,k+1} \\ y_{4,k+1} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} y_{1,k} \\ y_{2,k} \\ y_{3,k} \\ y_{4,k} \end{pmatrix} + \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ c_{41} & c_{42} & c_{43} \end{pmatrix} \begin{pmatrix} u_{1,k} \\ u_{2,k} \\ u^*_{trans} \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} \quad \text{(B.1)}$$

At steady state where $y_{i,k+1} = y_{i,k}$, the expression for y_1 in Eq.(B.1) for instance simplifies to

$$(1 - a_{11})y_{1,k} = a_{12}y_{2,k} + a_{13}y_{3,k} + c_{11}u_{1,k} + c_{12}u_{2,k} + c_{13}\,u_{2,k}^2 + d_1 \quad \text{(B.2)}$$

Substituting Eq.(B.2)and similar expressions for $y_{2,k}$ and $y_{3,k}$ into Eq.(3.46)yields the following:

$$\begin{aligned} &\left(\frac{n_{c,2_o}c_{21}}{1-a_{22}} + \frac{n_{c,3_o}c_{31}}{1-a_{33}}\right)\boldsymbol{y_{1,k}} + \left(\frac{n_{c,1_o}c_{12}}{1-a_{11}} + \frac{n_{c,3_o}c_{32}}{1-a_{33}}\right)\boldsymbol{y_{2,k}} \\ &+ \left(\frac{n_{c,1_o}c_{13}}{1-a_{11}} + \frac{n_{c,2_o}c_{23}}{1-a_{22}}\right)\boldsymbol{y_{3,k}} + \left(\frac{n_{c,1_o}c_{11}}{1-a_{11}} + \frac{n_{c,2_o}c_{21}}{1-a_{22}} + \frac{n_{c,3_o}c_{31}}{1-a_{33}} - n_{c,1_i}\right)\boldsymbol{u_1} \\ &+ \left(\frac{n_{c,1_o}c_{12}}{1-a_{11}} + \frac{n_{c,2_o}c_{22}}{1-a_{22}} + \frac{n_{c,3_o}c_{32}}{1-a_{33}} - n_{c,1_i}\right)\boldsymbol{u_2} \\ &+ \left(\frac{n_{c,1_o}c_{13}}{1-a_{11}} + \frac{n_{c,2_o}c_{23}}{1-a_{22}} + \frac{n_{c,3_o}3}{1-a_{33}}\right)\boldsymbol{u_2^2} + (n_{c,1_o}d_1 + n_{c,2_o}d_2 + n_{c,3_o}d_3) = 0 \end{aligned} \quad \text{(B.3)}$$

Setting the coefficient of each basis function and the constant term to zero yields the set of equality constraints for the exact satisfaction of the mass conservation law.

$$\begin{aligned} &\min_{\theta=\{\boldsymbol{a_{qp}},\boldsymbol{d_q}\}} \sum (y_{model} - y_{data})^2 \\ \boldsymbol{s.t} \quad &\frac{n_{c,2_o}c_{21}}{1-a_{22}} + \frac{n_{c,3_o}c_{31}}{1-a_{33}} = 0 \end{aligned} \quad \text{(B.4)}$$

$$\frac{n_{c,1_o}c_{12}}{1-a_{11}} + \frac{n_{c,3_o}c_{32}}{1-a_{33}} = 0$$

$$\frac{n_{c,1_o}c_{13}}{1-a_{11}} + \frac{n_{c,2_o}c_{23}}{1-a_{22}} = 0$$

$$\frac{n_{c,1_o}c_{11}}{1-a_{11}} + \frac{n_{c,2_o}c_{21}}{1-a_{22}} + \frac{n_{c,3_o}c_{31}}{1-a_{33}} - n_{c,1_i} = 0$$

$$\frac{n_{c,1_o}c_{12}}{1-a_{11}} + \frac{n_{c,2_o}c_{22}}{1-a_{22}} + \frac{n_{c,3_o}c_{32}}{1-a_{33}} - n_{c,1_i} = 0$$

$$\frac{n_{c,1_o}c_{13}}{1-a_{11}} + \frac{n_{c,2_o}c_{23}}{1-a_{22}} + \frac{n_{c,3_o}3}{1-a_{33}} = 0$$

$$n_{c,1_o}d_1 + n_{c,2_o}d_2 + n_{c,3_o}d_3 = 0$$

PART 2: With Hold-up Information

Case study: Isothermal CSTR

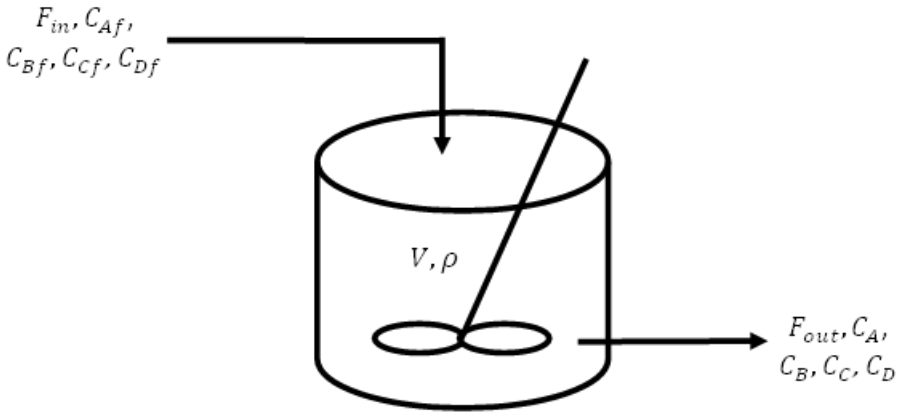


Fig. B. 7. Schematic diagram of isothermal CSTR.

Measured output variables:

$$y_1 = C_A, \qquad y_1 = C_B, \qquad y_1 = C_C, \qquad y_1 = C_D$$

Model:

The following is a general model for this system.

$$\begin{pmatrix} y_{1,k+1} \\ y_{2,k+1} \\ y_{3,k+1} \\ y_{4,k+1} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} y_{1,k} \\ y_{2,k} \\ y_{3,k} \\ y_{4,k} \end{pmatrix}$$

$$+ \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1nu} & c_{1*} \\ c_{21} & c_{22} & \dots & c_{2nu} & c_{2*} \\ c_{31} & c_{32} & \dots & c_{3nu} & c_{3*} \\ c_{41} & c_{42} & \dots & c_{4nu} & c_{4*} \end{pmatrix} \begin{pmatrix} u_{1,k} \\ u_{2,k} \\ \vdots \\ u_{nu,k} \\ u^*_{trans} \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} \tag{B.5}$$

In Eq.(B.5),nu is the number of changing input variables considered and the dimensions of u^*_{trans} and corresponding c_{i*} parameters depend on the number of additional basis functions selected by the BIDSAM algorithm.

Atom balance: $y_{j\ acc,k+1} = \sum F_{in} n_{j,p_i}(u_{p,k}) - \sum F_{out} n_{j,q_o}(y_{q,ddr,k+1})$ (B.6)

On substituting B.5 into B.6 and simplification, we have

$$y_{j\ acc\ k+1} = -F_{out}\begin{pmatrix}\sum_{q=1}^{4} n_{j,q_o}a_{q1}\\ \sum_{q=1}^{4} n_{j,q_o}a_{q2}\\ \sum_{q=1}^{4} n_{j,q_o}a_{q3}\\ \sum_{q=1}^{4} n_{j,q_o}a_{q4}\end{pmatrix}^T\begin{pmatrix}y_{1,k}\\ y_{2,k}\\ y_{3,k}\\ y_{4,k}\end{pmatrix} + \begin{pmatrix}F_{in}n_{j,1_i} - F_{out}\sum_{q=1}^{4} n_{j,q_o}c_{q1}\\ F_{in}n_{j,2_i} - F_{out}\sum_{q=1}^{4} n_{j,q_o}c_{q2}\\ \vdots\\ \vdots\\ F_{in}n_{j,nu_i} - F_{out}\sum_{q=1}^{4} n_{j,q_o}c_{q,nu}\\ -F_{out}\sum_{q=1}^{4} n_{j,q_o}c_{q*}\end{pmatrix}^T\begin{pmatrix}u_{1,k}\\ u_{2,k}\\ \vdots\\ \vdots\\ u_{nu,k}\\ u^*_{trans}\end{pmatrix} + \sum_{q=1}^{4} n_{j,q_o}d_q \quad \text{(B.7)}$$

Case 1: $F_{in} = F_{out} = constant$

$$u_1 = C_{Af},\ u_2 = C_{Bf},\ u_3 = C_{Cf},\ u_4 = C_{Df}$$

$$y_{j\ acc\ k+1} = -F_{out}\begin{pmatrix}\sum_{q=1}^{4} n_{j,q_o}a_{q1}\\ \sum_{q=1}^{4} n_{j,q_o}a_{q2}\\ \sum_{q=1}^{4} n_{j,q_o}a_{q3}\\ \sum_{q=1}^{4} n_{j,q_o}a_{q4}\end{pmatrix}^T\begin{pmatrix}y_{1,k}\\ y_{2,k}\\ y_{3,k}\\ y_{4,k}\end{pmatrix} + \begin{pmatrix}F_{in}n_{j,1_i} - F_{out}\sum_{q=1}^{4} n_{j,q_o}c_{q1}\\ F_{in}n_{j,2_i} - F_{out}\sum_{q=1}^{4} n_{j,q_o}c_{q2}\\ F_{in}n_{j,3_i} - F_{out}\sum_{q=1}^{4} n_{j,q_o}c_{q3}\\ F_{in}n_{j,4_i} - F_{out}\sum_{q=1}^{4} n_{j,q_o}c_{q,4}\\ -F_{out}\sum_{q=1}^{4} n_{j,q_o}c_{q*}\end{pmatrix}^T\begin{pmatrix}\boldsymbol{u_{1,k}}\\ \boldsymbol{u_{2,k}}\\ \boldsymbol{u_{3,k}}\\ \boldsymbol{u_{4,k}}\\ \boldsymbol{u^*_{trans}}\end{pmatrix} \quad \text{(B.8)}$$

$$+\sum_{q=1}^{4} n_{j,q_o}d_q$$

Case 2: *F_{out} is an input variable u_1*

$$u_1 = F_{out},\ u_2 = C_{Af},\ u_3 = C_{Bf},\ u_4 = C_{Cf},\ u_5 = C_{Df}$$

$$y_{j\ acc\ k+1} = -\begin{pmatrix} \sum_{q=1}^{4} n_{j,q_o} a_{q1} \\ \sum_{q=1}^{4} n_{j,q_o} a_{q2} \\ \sum_{q=1}^{4} n_{j,q_o} a_{q3} \\ \sum_{q=1}^{4} n_{j,q_o} a_{q4} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{y_{1,k}} \\ \boldsymbol{y_{2,k}} \\ \boldsymbol{y_{3,k}} \\ \boldsymbol{y_{4,k}} \end{pmatrix} \boldsymbol{u_{1,k+1}} + \begin{pmatrix} F_{in} n_{j,1_i} \\ F_{in} n_{j,2_i} \\ F_{in} n_{j,3_i} \\ F_{in} n_{j,4_i} \\ F_{in} n_{j,5_i} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{u_{1,k}} \\ \boldsymbol{u_{2,k}} \\ \boldsymbol{u_{3,k}} \\ \boldsymbol{u_{4,k}} \\ \boldsymbol{u_{5,k}} \end{pmatrix}$$

$$+ \left(\begin{pmatrix} -\sum_{q=1}^{4} n_{j,q_o} c_{q1} \\ -\sum_{q=1}^{4} n_{j,q_o} c_{q2} \\ -\sum_{q=1}^{4} n_{j,q_o} c_{q3} \\ -\sum_{q=1}^{4} n_{j,q_o} c_{q,4} \\ -\sum_{q=1}^{4} n_{j,q_o} c_{q,5} \\ -\sum_{q=1}^{4} n_{j,q_o} c_{q*} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{u_{1,k}} \\ \boldsymbol{u_{2,k}} \\ \boldsymbol{u_{2,k}} \\ \boldsymbol{u_{2,k}} \\ \boldsymbol{u_{2,k}} \\ \boldsymbol{u_{5,k}} \\ \boldsymbol{u^*_{trans}} \end{pmatrix} + \sum_{q=1}^{4} n_{j,q_o} d_q \right) \boldsymbol{u_{1,k+1}} \quad \text{(B.9)}$$

*Note that the prediction of $y_{j\ acc\ k+1}$ requires $u_{1,k+1}$ but this is not an issue since this is required only for the training data.

Case 3: *F_{in} and F_{out} are input variables*

$$u_1 = F_{in},\ u_2 = F_{out},\ u_3 = C_{Af},\ u_4 = C_{Bf},\ u_5 = C_{Cf},\ u_6 = C_{Df}$$

$$y_{j\ acc\ k+1} = -\begin{pmatrix} \sum_{q=1}^{4} n_{j,q_o} a_{q1} \\ \sum_{q=1}^{4} n_{j,q_o} a_{q2} \\ \sum_{q=1}^{4} n_{j,q_o} a_{q3} \\ \sum_{q=1}^{4} n_{j,q_o} a_{q4} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{y_{1,k}} \\ \boldsymbol{y_{2,k}} \\ \boldsymbol{y_{3,k}} \\ \boldsymbol{y_{4,k}} \end{pmatrix} \boldsymbol{u_{1,k+1}} + \begin{pmatrix} \boldsymbol{u_{2,k}} n_{j,1_i} - \boldsymbol{u_{1,k+1}} \sum_{q=1}^{4} n_{j,q_o} c_{q1} \\ \boldsymbol{u_{2,k}} n_{j,2_i} - \boldsymbol{u_{1,k+1}} \sum_{q=1}^{4} n_{j,q_o} c_{q2} \\ \vdots \\ \vdots \\ \boldsymbol{u_{2,k}} n_{j,nu_i} - \boldsymbol{u_{1,k+1}} \sum_{q=1}^{4} n_{j,q_o} c_{q,6} \\ -\boldsymbol{u_{1,k+1}} \sum_{q=1}^{4} n_{j,q_o} c_{q*} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{u_{1,k}} \\ \boldsymbol{u_{2,k}} \\ \vdots \\ \vdots \\ \boldsymbol{u_{6,k}} \\ \boldsymbol{u^*_{trans}} \end{pmatrix} + \sum_{q=1}^{4} n_{j,q_o} d_q \qquad \text{(B.10)}$$

Validation Plots for Steady State Modeling of Absorber Unit of CO_2 Capture Plant

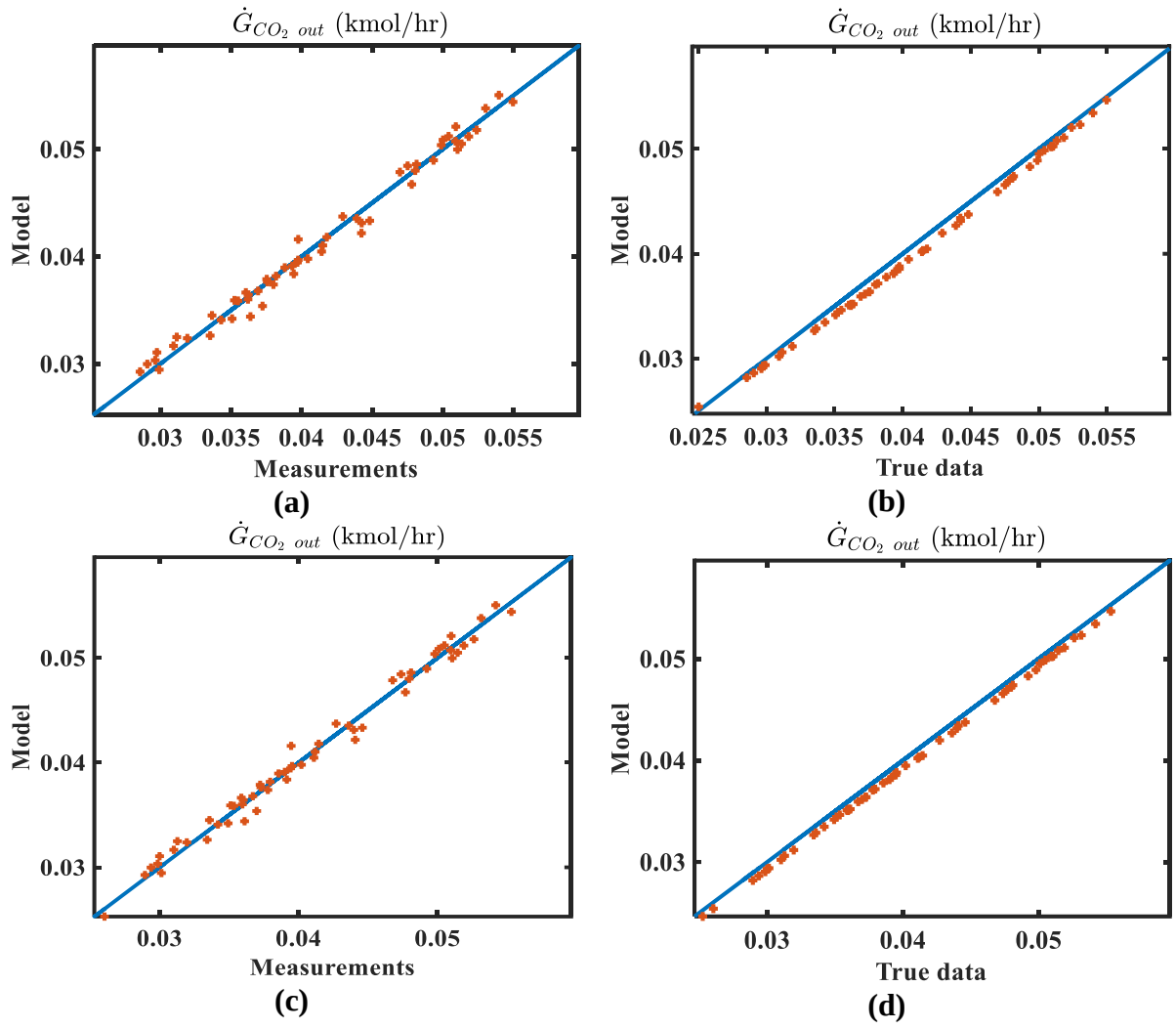


Fig. B. 8. Validation plots for model developed from noisy data using Algorithm II: (a)-(b) without mass constraints, (c)-(d) with mass constraint.

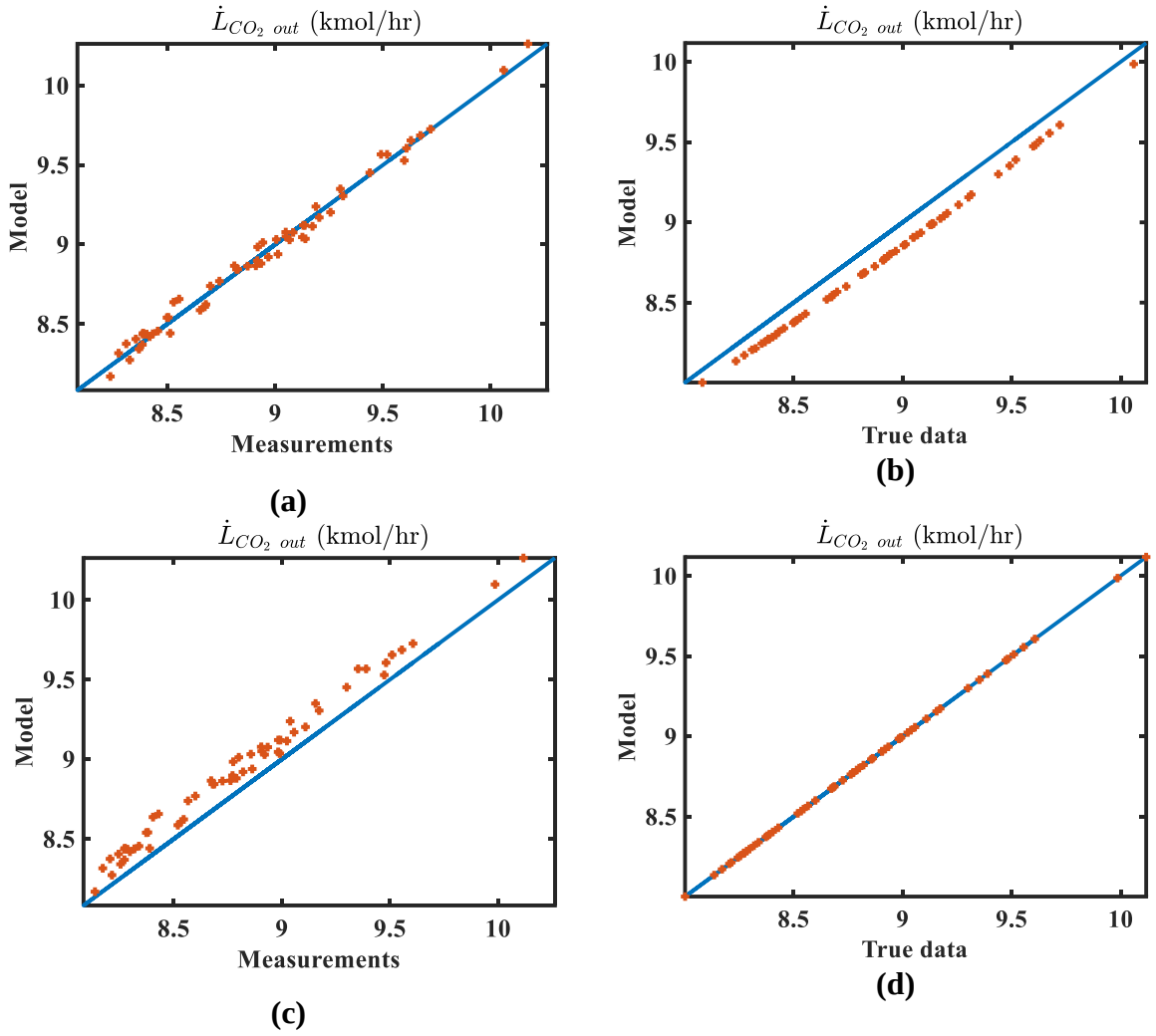


Fig. B. 9. Validation plots for model developed from noisy data using Algorithm I: (a)-(b) without mass constraints, (c)-(d) with mass constraint.

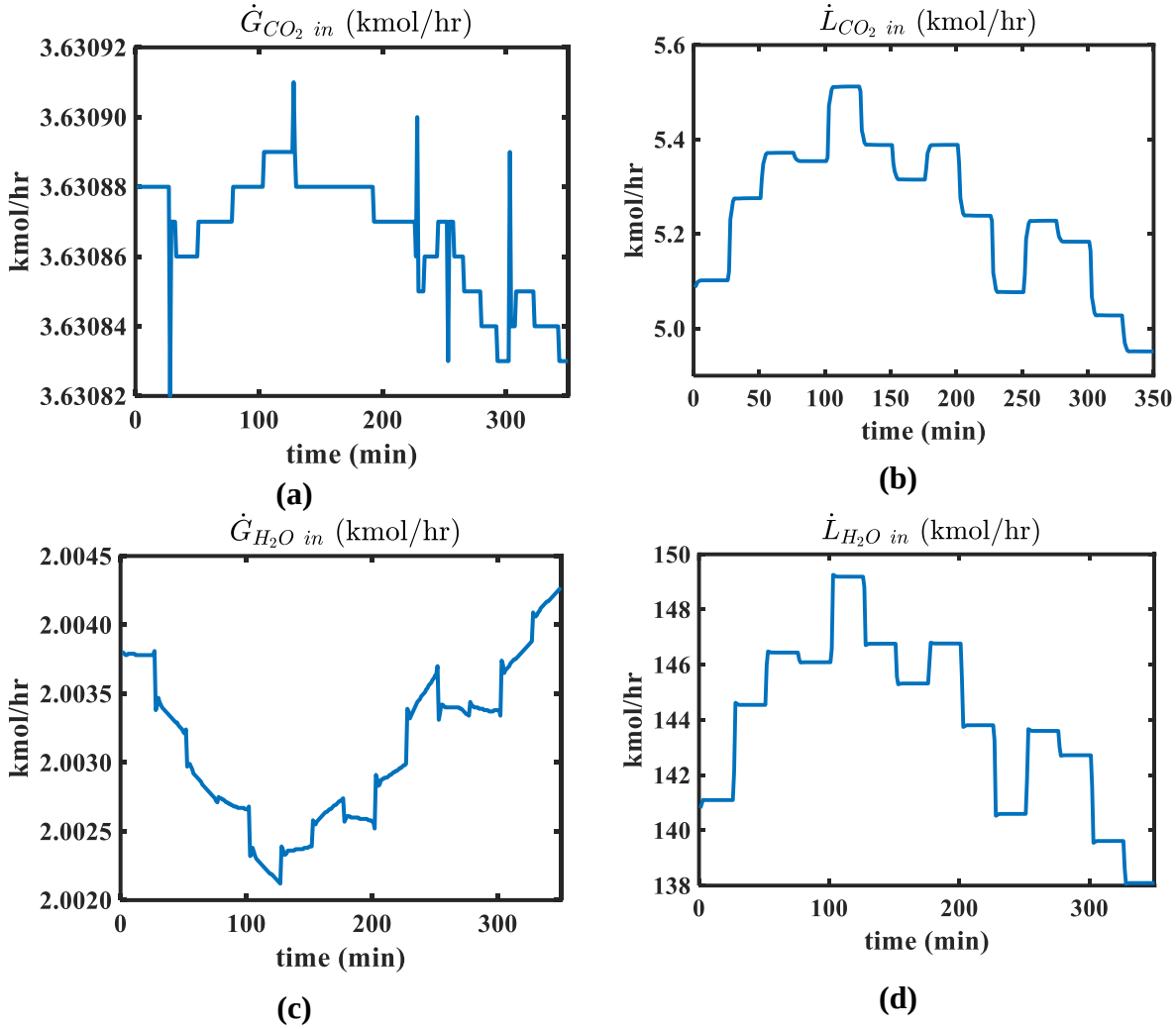


Fig. B. 10. Input variables for validation data.

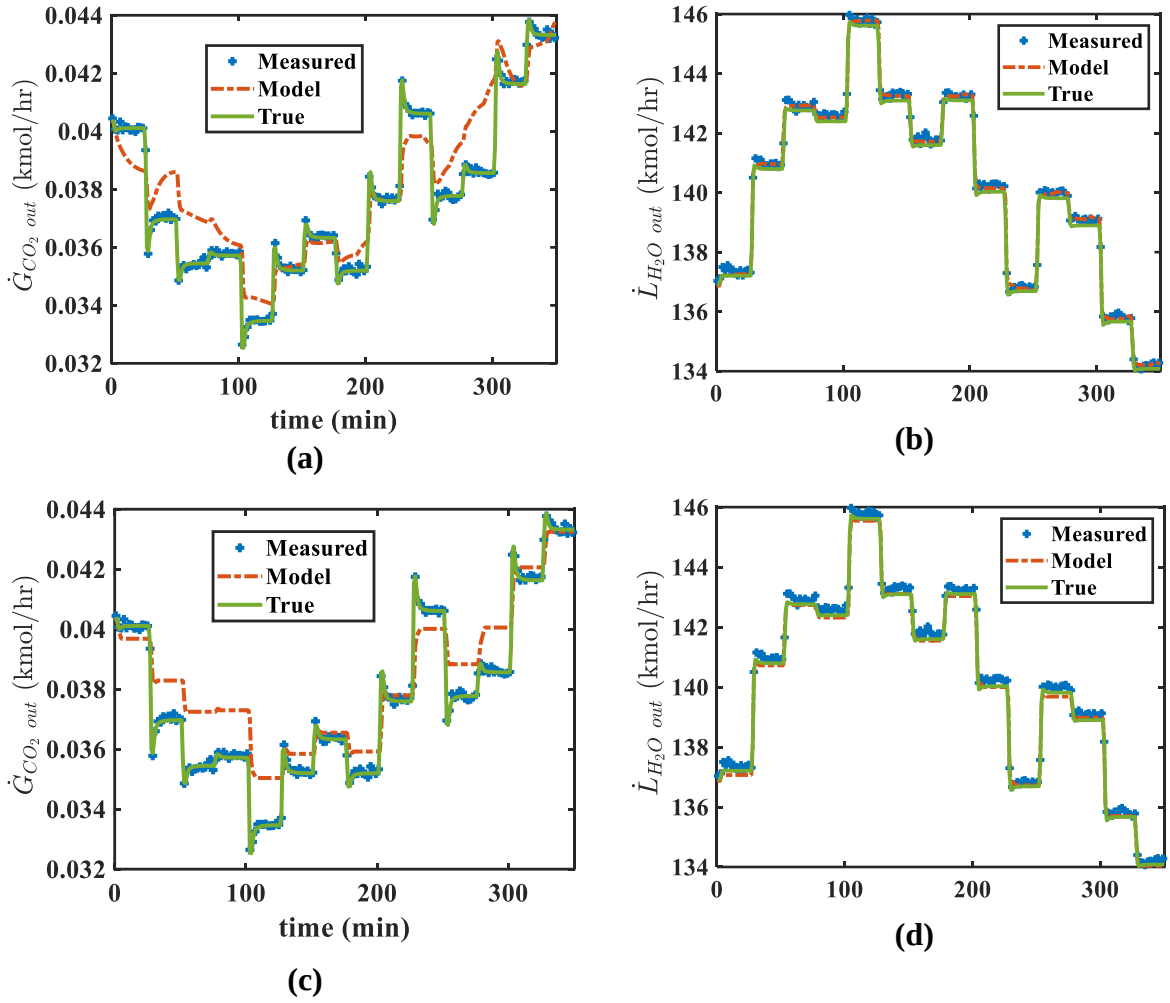


Fig. B. 11. Validation plots for model developed from noisy data using Algorithm I: (a)-(b) without mass constraints, (c)-(d) with mass constraint.

Derivation of constraints for a constant volume reactor

Consider the CSTR shown in Fig. B. 7 with the model Eq.(B.5). The following atom balance is carried out assuming a sampling time of 1unit time:

Atom balance:

$$\sum n_{j,q_o}(Vy_{q,k+1} - Vy_{q,ddr,k}) = \sum F_{in} n_{j,p_i}(u_{p,k}) - \sum F_{out} n_{j,q_o}(y_{q,k+1}) \qquad \text{(B.11)}$$

At constant volume, V is factorized out of the LHS of Eq.(B.11) and the respective expressions for $y_{q,k+1}$ are substituted into Eq.(B.11). On simplification, the following coefficients of the respective basis function give the desired constraints to be enforced for the satisfaction of conservation laws.

$$y_{q,k} \qquad V\left(\sum_{q\neq r}\left(n_{j,q_o}a_{qr}\right) + n_{j,q_o}(a_{qq}-1)\right) + F_{out}\sum n_{j,q_o}a_{qr}$$

For $q, r = 1 \ldots ny$

$$u_{p,k} \qquad (V + F_{out})\left(\sum_{q\neq r}\left(n_{j,q_o}c_{qr}\right)\right) - F_{in}n_{j,p_i}$$

For $q, r = 1 \ldots ny$

$p = 1 \ldots nu$ (B.12)

$$u^*_{trans,k} \qquad (V + F_{out})\left(\sum_{q\neq r}\left(n_{j,q_o}c_{qr}\right)\right)$$

For $q, r = 1 \ldots ny$

$p = 1 \ldots ntrans$

Constant $\qquad (V + F_{out})\left(\sum_{q\neq r} d_i\right)$

Appendix C: Additional results for the development and validation of the Neural Network and Gaussian RBF Models

Additional Results from Hybrid All-Nonlinear NNs

This section contains additional results obtained during modeling of complex nonlinear dynamic chemical process systems using hybrid series and parallel all-nonlinear static-dynamic neural network models.

Table C. 1. Training time for different solution combinations for NLS-NLD, NLD-NLS and NLS||NLD for the NCCC pilot plant for CO_2 capture

Static / Dynamic	Training Time (sec) for NLS-NLD				Training Time (sec) for NLD-NLS				Training Time (sec) for NLS||NLD			
	GD	SCG	LM	BFGS	GD	SCG	LM	BFGS	GD	SCG	LM	BFGS
GD	98	72	80	68	126	91	85	92	146	101	96	88
SCG	92	50	54	56	120	70	61	84	128	90	87	92
LM	96	52	64	65	116	68	73	71	152	92	114	86
BFGS	93	49	60	48	110	64	66	82	130	90	98	93

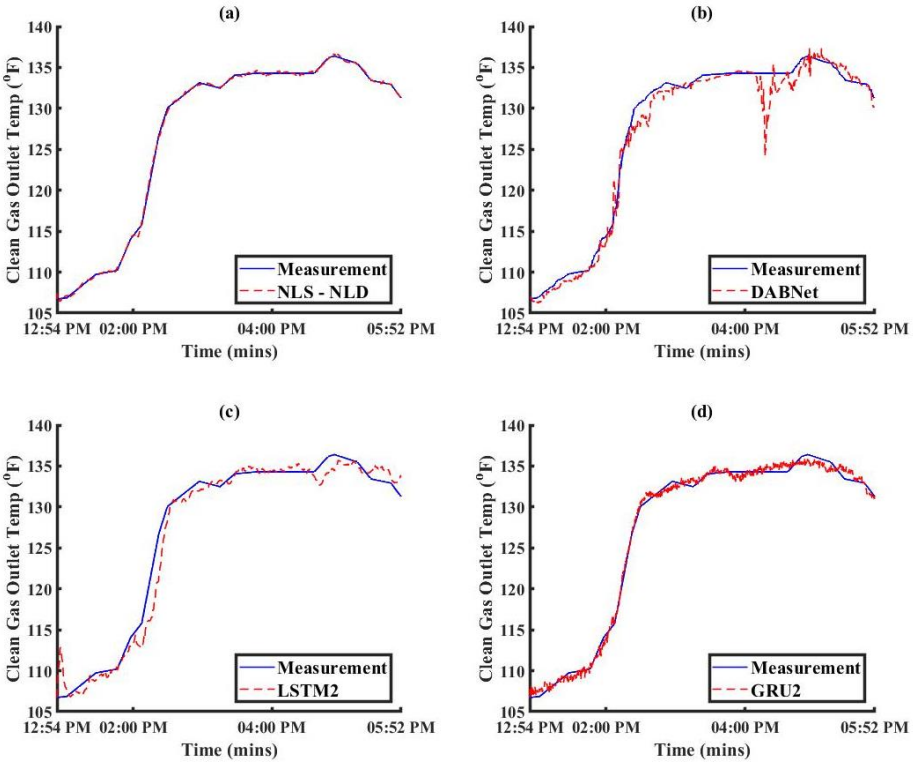


Fig. C. 1. Comparison between the (a) NLS-NLD, (b) DABNet, (c) LSTM2, and (d) GRU2 for the validation data for the flue gas outlet temperature

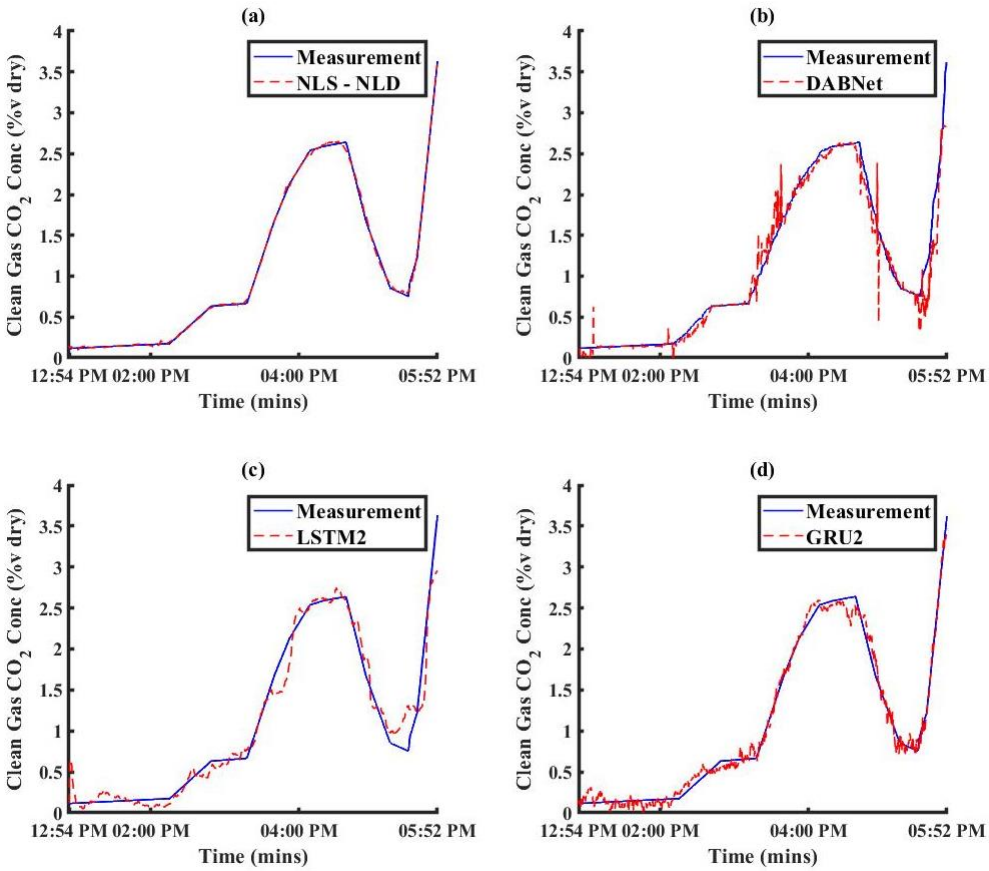


Fig. C. 2. Comparison between the (a) NLS-NLD, (b) DABNet, (c) LSTM2, and (d) GRU2 for the validation data for CO_2 concentration in the outlet flue gas stream

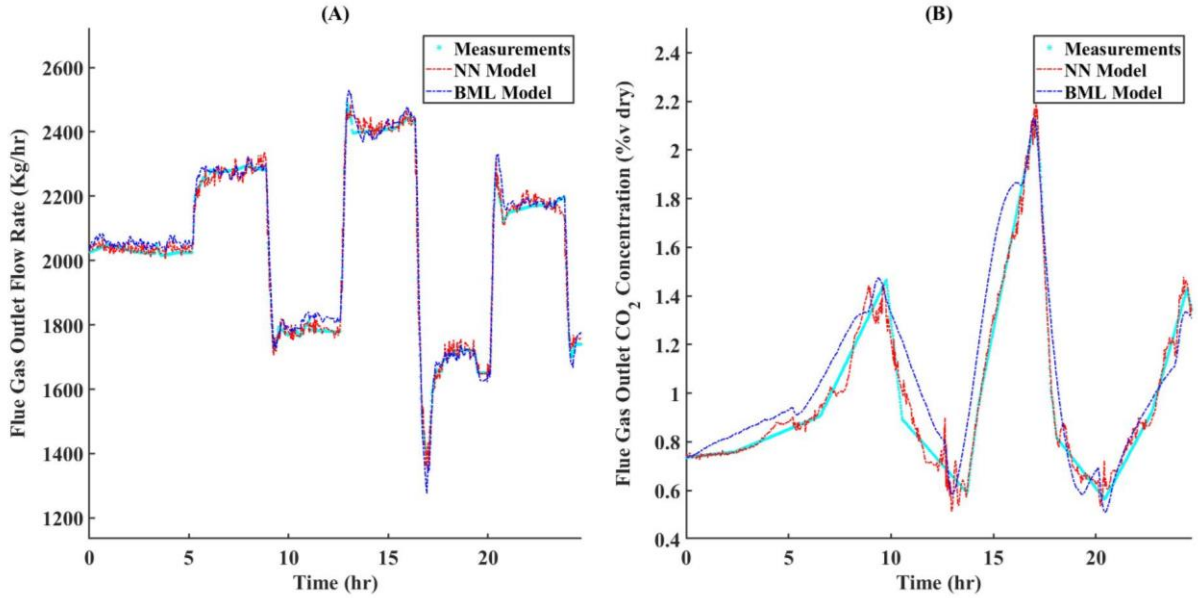


Fig. C. 3. Comparison of prediction results obtained from NN and BML models for (A) flue gas outlet flow rate and (B) flue gas outlet CO_2 concentration

Table C. 2. Optimal parameter values of NN and BML models for the CO_2 Capture Pilot Plant

- Optimal NN (NLS || NLD) Model

$$\boldsymbol{y}(t) = \boldsymbol{y_{stat}}(t) + \boldsymbol{y_{dyn}}(t)$$

$$\boldsymbol{y_{stat}}(t) = \phi_{o,s}\left[W_{out,s}\,\phi_{h,s}\left(W_{hid,s}\boldsymbol{u_{NN}}(t) + \boldsymbol{\beta_{h,s}}\right) + \boldsymbol{\beta_{o,s}}\right]$$

$$\boldsymbol{y_{dyn}}(t) = \phi_{o,d}\left[W_{out,d}\,\phi_{h,d}\left(W_{hid,d}\boldsymbol{u_{NN}}(t) + \boldsymbol{\beta_{h,d}} + W_{fb}\boldsymbol{y_{dyn}}(t-1)\right) + \boldsymbol{\beta_{o,d}}\right]$$

$$\phi_{h,s}(\boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{x})},\quad \phi_{o,s}(\boldsymbol{x}) = \boldsymbol{x}$$

$$\phi_{h,d}(\boldsymbol{x}) = \frac{\exp(\boldsymbol{x}) - \exp(-\boldsymbol{x})}{\exp(\boldsymbol{x}) + \exp(-\boldsymbol{x})},\quad \phi_{o,d}(\boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{x})}$$

$$W_{hid,s} = \begin{bmatrix} 3.80 & -0.75 & 3.12 & 1.37 & -13.13 \\ -0.10 & 0.23 & 1.87 & 10.27 & 0.76 \\ 2.11 & -0.28 & 1.42 & -1.82 & -9.16 \\ 22.24 & 0.14 & -2.37 & 4.12 & -6.14 \\ 0.19 & -0.003 & 0.05 & -0.06 & -0.11 \end{bmatrix}^T$$

$$W_{out,s} = \begin{bmatrix} 1.12 & 6.99 & 2.16 & -1.32 & -0.52 \\ 0.22 & -3.98 & 0.26 & 0.17 & -0.19 \\ 1.06 & 10.70 & 2.22 & -1.85 & 1.52 \end{bmatrix}$$

$$\boldsymbol{\beta_{h,s}} = [17.36 \quad 1.15 \quad -0.78 \quad -4.20 \quad -14.52]^T,\ \boldsymbol{\beta_{o,s}} = [-6.35 \quad 2.60 \quad -9.44]^T$$

$$W_{hid,d} = \begin{bmatrix} 0.48 & -5.96 & 5.78 & -14.01 & 0.50 \\ 0.90 & -24.61 & -1.32 & 18.58 & -2.07 \end{bmatrix}$$

$$W_{out,d} = \begin{bmatrix} 15.30 & -0.60 \\ -24.35 & -6.67 \\ -5.44 & 3.25 \end{bmatrix},\ W_{fb} = \begin{bmatrix} 8.96 & -2.68 & -6.57 \\ -11.03 & 2.25 & -1.49 \end{bmatrix}$$

$$\boldsymbol{\beta_{h,d}} = [18.03 \quad 15.87]^T,\ \boldsymbol{\beta_{o,d}} = [-16.14 \quad -16.36 \quad -9.26]^T$$

- Optimal BML Model

$$y_k = \boldsymbol{A}y_{k-1} + \boldsymbol{B}U^*_{k-1}$$

$$A = \begin{pmatrix} 0.99 & 0.18 & -0.03 \\ -0.04 & 0.67 & 0.06 \\ 0.01 & 0.02 & 1.02 \end{pmatrix}$$

$$B = \begin{pmatrix} -0.02 & 1.25 & 2.34 & 4e-3 & 5e-3 & -0.33 & -0.82 & -0.03 & -1e-4 \\ 0.04 & -2.79 & -5.53 & 1e-3 & -0.01 & 0.66 & 2.19 & -0.03 & 3e-4 \\ 3e-3 & 0.75 & 1.38 & -6e-4 & -0.02 & -0.17 & -0.55 & 0.05 & -3e-6 \end{pmatrix}$$

Additional Results and Algorithms for METCNN Models

This section includes the algorithmic flowchart showing the training algorithm for dynamic METCNN models as well additional training / simulation results obtained from the optimal METCNN / MECNN models.

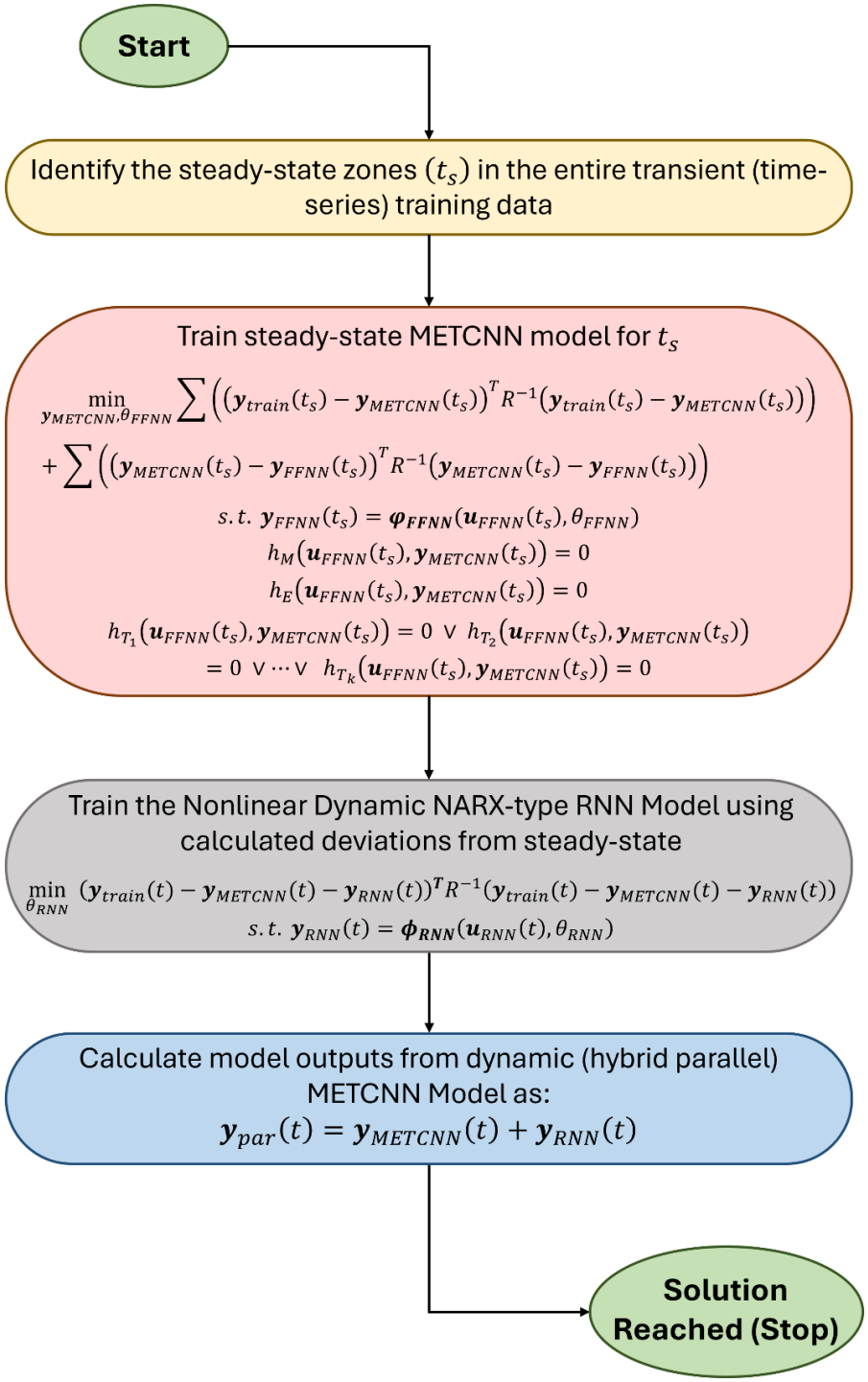


Fig. C. 4. Training algorithm for dynamic (hybrid parallel) METCNN model in absence of system holdup measurements

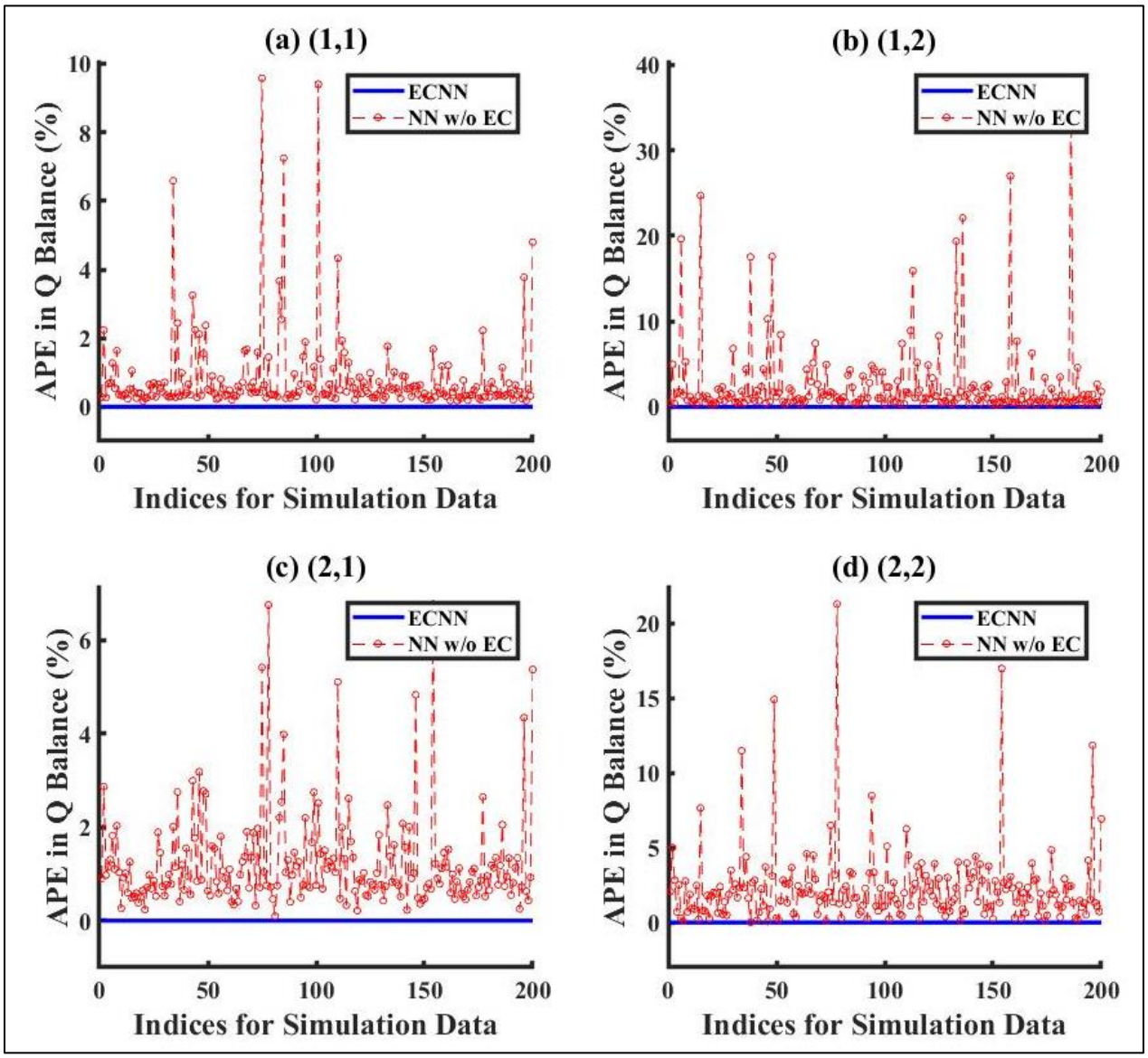


Fig. C. 5. Comparison of error between ECNN and NN w/o energy constraints for energy balance at intermediate grids during simulation

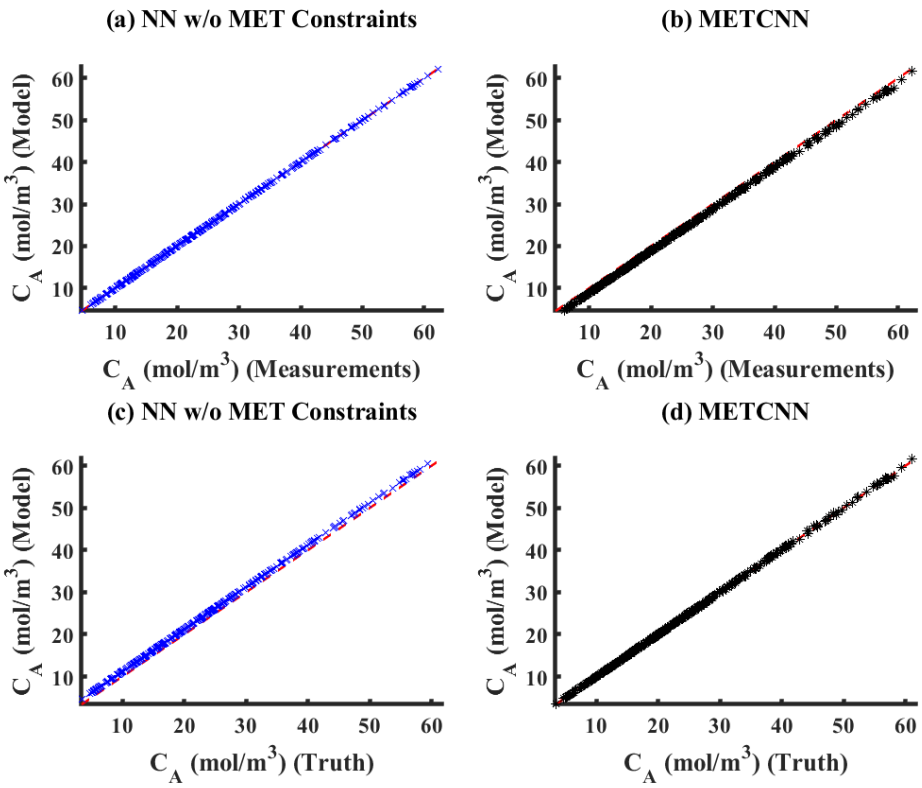


Fig. C. 6. Comparison of results between METCNN and NN w/o METC for simulation data of outlet concentration of A (C_A)

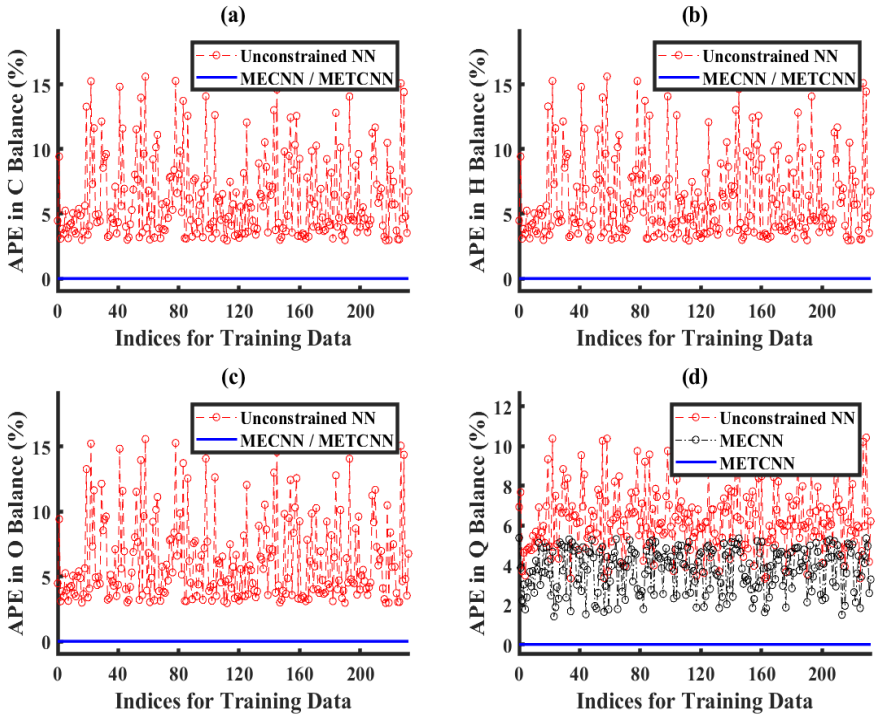


Fig. C. 7. Comparison between METCNN, MECNN and NN w/o METC in terms of violating (a) carbon (C), (b) hydrogen (H), (c) oxygen (O) and (d) energy (Q) balance constraints during training

Additional Results from RBF / GRAB-NN Models

This section includes additional training / validation results obtained from the optimal data-driven models for different case studies considered in this work.

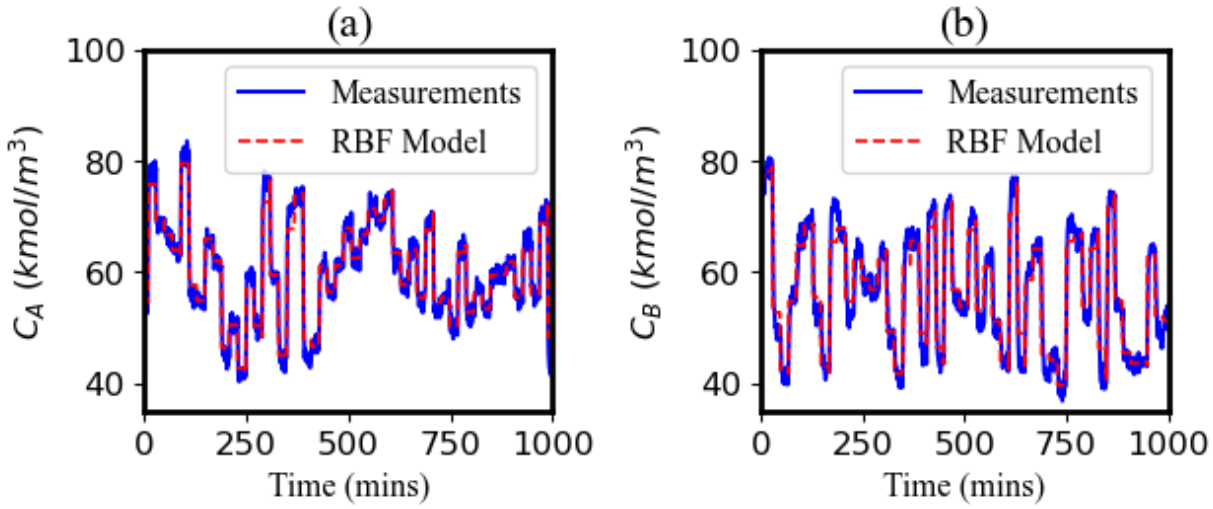


Fig. C. 8. Results from the optimal unconstrained RBF model with deterministic centers obtained through the approximate training algorithm for validation data of (a) C_A and (b) C_B

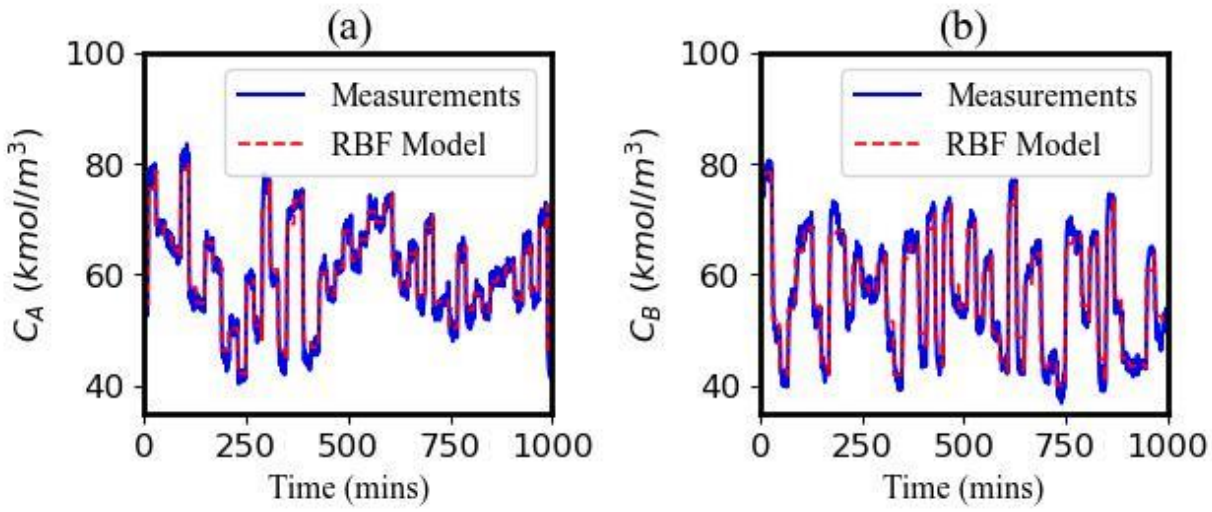


Fig. C. 9. Results from the optimal unconstrained RBF model with probabilistic centers obtained through the approximate training algorithm for validation data of (a) C_A and (b) C_B

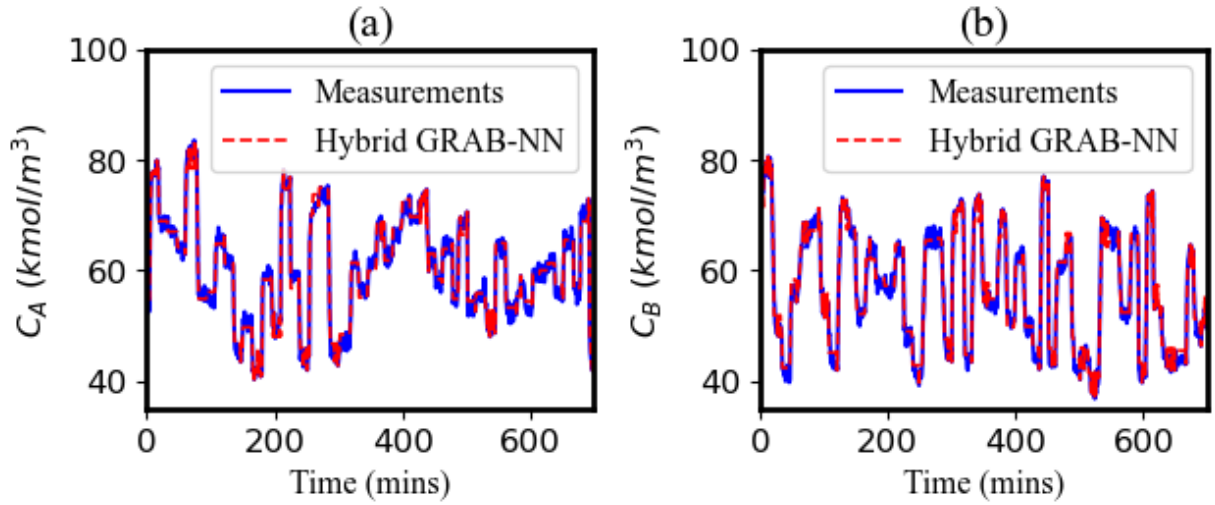


Fig. C. 10. Results from optimal hybrid GRAB-NN model for validation data of (a) C_A and (b) C_B

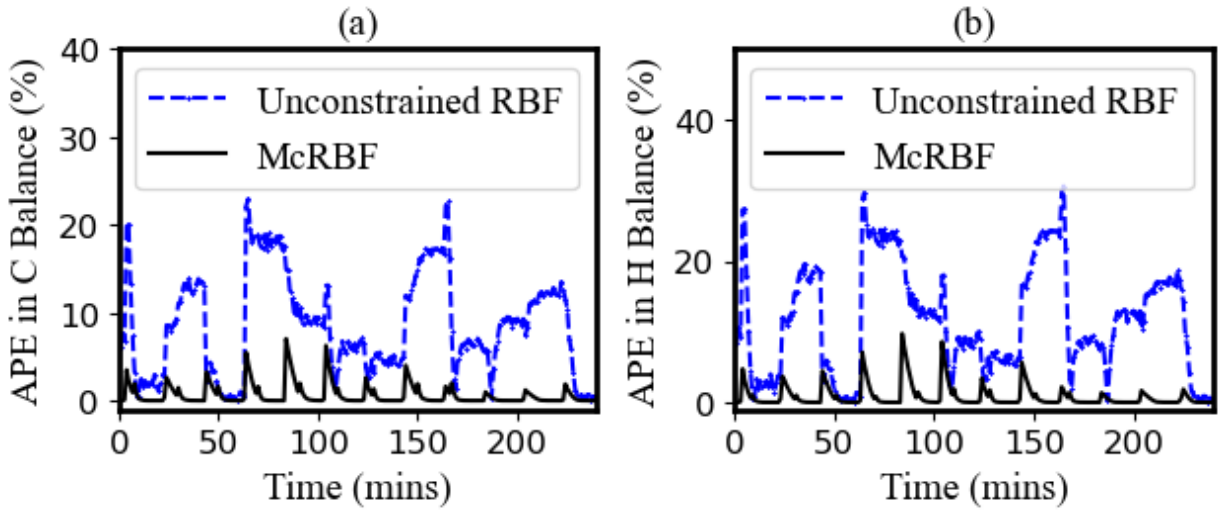


Fig. C. 11. Comparison between dynamic McRBF and unconstrained RBF models in terms of violating (a) carbon (C) and (b) hydrogen (H) atom balance constraints during validation

Appendix D : Superheater Damage Evaluation Software Specification Document

➢ Introduction

- **Purpose**

Creep damage, caused by long-term exposure to high temperatures, has traditionally been estimated under static conditions rather than through online, time-varying signals that capture cumulative damage. With power plants shifting towards more flexible operations, more adaptive and predictive approaches are needed. This specification establishes a foundation for assessing creep and fatigue damage using available time series data, providing life consumption estimates at sensor locations. This software is designed such that the WVU model could be integrated in a future work, when that model becomes more readily deployable at scale. A full description of that model can be found in Section 5.

This document outlines a specification to develop software to evaluate superheater health by assessing creep and fatigue life in power-generating boiler systems. This will be done directly using available plant data. The results are intended to be incorporated into an asset management system to better prepare for maintenance of superheater systems. This intent is to move the power industry towards more robust creep and fatigue assessment tools. This software is designed to assess damage incrementally by fully considering time series data.
This specification includes analyses to identify high-risk operations from existing sensor data, estimate fatigue life consumption by calculating observed stresses, and assess creep life consumption by estimating time spent at creep-inducing conditions. This software enhances the ability to assess superheater health in response to the fluctuating conditions of modern power plant operations.

Additionally, this document establishes a roadmap for partners, such as EPRI (Electric Power Research Institute), to develop a modular framework that can accommodate additional analyses over time. It describes the data storage and processing requirements to support this framework, along with the core analyses and their associated data and metadata needs for the assessment of creep and fatigue life for superheater systems. This document defines the requirements for evaluating temperature, load, flow rate, and pressure cycles under various loading conditions, and the geometric and material properties required to extend these assessments to full creep and fatigue damages assessments. It also includes requirements for integrating an existing oxidation thickness calculator and conducting a Weibull assessment of header stub failure rates. Details on these requirements are found in Section

Analysis Requirements. This adaptable framework supports adding future analyses and is designed to be reliable for different types of heat exchanger equipment, including Heat Recovery Steam Generators (HRSGs).

To achieve this broad applicability, the analyses are intended to be general calculations without specific assumptions about the heat exchanger system to be analyzed. The data, however limited or abundant, and the geometry provided will determine how the results should be interpreted for that system. The developed software will clearly indicate any deviations from this assumption, where a particular analysis is *only* applicable to a particular type of boiler system.
This document aims to ensure that developers, engineers, and end-users understand the system's capabilities, the technology's potential impact, and the steps required for successful implementation. It defines the requirements for building software to deploy superheater health assessment tools, including secure data transfer and storage, analysis requirements, and clear presentation of results. These guidelines should be followed when necessary time and resources for development are available.

- **Scope**

This document outlines the scope of the Superheater Damage Evaluation Software project, aimed at developing a framework of tools for superheater damage assessment. This framework defines the tools required to assess creep and fatigue life in steam boiler superheater headers and tubing using time series data from currently installed sensors on superheater hardware.

Key Aspects of the Project Scope Include:

Model Development and Validation: Define a framework of tools that allow for the analysis of creep and fatigue life of timeseries data. Develop a software framework that allows for modular addition of new tools that allows for easy integration of updated models over time.

Technology Validation and Maturation: Conduct rigorous software testing, validation, and integration across all software components. Verify accuracy of results, assess robustness and scalability for different use cases, and guarantee adaptability to different boiler types, including HRSGs.

Stakeholder Collaboration: Collaborate with project stakeholders such as EPRI, power plant operators, and other industry partners. Define the end state of the evaluation tools, understanding use cases, and integrate stakeholder feedback to meet the practical needs of the industry.

User Interface and Experience: Design an intuitive and accessible user interface for plant operators and engineers. Facilitate easy monitoring, data analysis, and reporting to support decision-making related to superheater maintenance and operation.

Documentation and Training: Provide comprehensive documentation and training materials including user manuals, troubleshooting guides, and training programs.

Futureproofing and Scalability: Consider future enhancements and scalability of the system to accommodate advances in technology.

This document provides a detailed roadmap that outlines the steps necessary to achieve these objectives, ensuring that the Superheater Damage Evaluation Software evolves into a valuable tool for enhancing the safety, efficiency, and longevity of power plant superheaters.

- **References**

This section lists the documents and resources directly referenced in the development of this specification document for the Superheater Damage Evaluation Software. These references outline key methodologies, project goals, and collaboration details.

Key References

- EPRI Member / Stakeholder Superheater Damage Evaluation Software Survey Responses (Section 6)

- **Document Overview**

This specification document guides the development of a Superheater Damage Evaluation Software. It outlines each phase of the system's design, development, and deployment. Below is an overview of the following sections contained in this document:

Stakeholder Needs: Discusses requirements identified through stakeholder interviews, highlighting the requirements for the software that the Superheater Damage Evaluation Software aims to fulfill.

Functional Requirements: Details the software's required functionalities, including data connectivity, storage, and tasks the software must perform. Describes how the software will be used and shows example mock-ups of different screens.

Non-Functional Requirements: Specifies performance metrics such as processing capabilities and reporting functionalities. It also covers data privacy, protection, and transmission.

Analysis Requirements: Lists analyses that the software must perform, including requirements for each analysis. For each boiler type, specifies what data, geometry, and material properties are required.

User Requirements: Defines configuration options for each user to define software behavior for their specific superheater system.

System Requirements: Explains how data is collected, stored, and maintained, ensuring scalability as datasets grow over time.

Quality Attributes: Ensures data completeness and accuracy of datasets upon upload to the software. This section also covers how the analyses are validated and how users provide feedback on the analyses provided.

Implementation Roadmap: Outlines the implementation phases, providing a timeline and description of the development, testing, and deployment stages.

This document serves as a roadmap and a guide for developers, architects, and project managers throughout the development lifecycle of the Superheater Damage Evaluation Software. It ensures alignment with the strategic goals established and provides a clear framework for software development.

➢ Stakeholder Needs

The development of the Superheater Damage Evaluation Software is driven by specific needs identified through discussions with key stakeholders, including power plant operators and maintenance teams. These needs shape the system's functionality, ensuring effectiveness and user acceptance. This section details the core requirements and expectations expressed by the stakeholders.

Reactivity vs. Proactivity

Current Practices: Stakeholders primarily rely on inspections to assess the state of headers and tubing. As coal boilers age and reach end of life, the focus has shifted towards a more reactive approach, rather than spending O&M budgets on damage prevention. In contrast, HRSGs have lower in-service experience and present inspection challenges due to limited access, which makes it difficult to predict where failures may occur.

Desired Change: There is a desire for more simple tools that help users estimate damage and understand the associated operational risk to specific hardware components. For HRSGs, there is a desire for tools that will aid in locating areas of risk in superheater components, especially where inspections are difficult to perform.

Data Availability and Usage

Current Data Availability: Many stakeholders primarily base maintenance decisions on inspection results and other non-destructive testing methods. While inspection results are stored, damage is not tracked in between inspections. Installed thermocouples are rare, and when available, they are not used to assess component condition.

Basic vs. Advanced Analysis: Stakeholders prefer basic, standard analyses over complex, real-time analytics. For example, they highlighted the need to identify high-temperature excursions likely to cause significant damage, focusing on practical, immediate insights that guide maintenance decisions. Any third-party tools need to account for protecting data and maintaining user privacy.

Tool Functionality and Accessibility

Web-Based Solutions: There is a moderate interest in adopting web-based tools for broader access and easier data management. This includes functionalities where users can simply upload data to the tool, moving away from traditional, manual systems.

Data Upload and Analysis: Stakeholders envision a system where data can be easily uploaded and automatically analyzed to determine the appropriate analysis. This includes the ability to map tags and specify units, simplifying the transition from data collection to analysis.

Modular and Informative Analysis Model

Core Model Concept: Stakeholders are interested in a core "model" within the system that performs different analyses based on uploaded data. This model should adapt its analyses based on the quantity and type of data provided. For example, if the software has previously been used to analyze a superheater and new sensor data becomes available, the software should be able to extend any analyses to those new sensors.

Suggestive Analytics: Stakeholders indicated that the number of subject matter experts within their companies has been trending downward. The software should clearly inform the user of what data is required for each analysis to ensure meaningful insights. The software should be straightforward and simple to use, minimizing the time and effort from staff to operate it.

User Interface and Result Display

Functional Interface: The system should feature an interface that allows users to easily upload files and see what analyses can be performed with the available data.

Results Display: After analysis selection, the GUI guides the user through mapping their data and defining their superheater system. Stakeholders expressed the need for a "display results" template that consistently shows analysis results in a user-friendly format.
These identified needs underline the importance of developing a system that is technically capable and user centered. The Superheater Damage Evaluation Software aims to meet these needs by being simple to use, providing meaningful damage risk assessments, and aiding in localizing those risks to specific areas for further inspections.

➢ Functional Requirements

This section of the document outlines the essential functional requirements for the Superheater Damage Evaluation Software. These requirements ensure the system performs its intended functions correctly and meets the needs identified in the stakeholder discussions. The following are the primary functionalities that the system must support to achieve the goals of proactive monitoring and efficient data analysis.

- **Software Overview**

The software will be engineered to evaluate creep and fatigue damage in superheater systems from sensor data uploaded by users. It will offer a suite of functionalities, guiding users through identifying their analysis options, mapping data channels to a three-dimensional superheater model, performing detailed data analyses, and generating insightful reports. These reports should include identifying timeframes where risk of creep and fatigue damage were particularly high, tabulating fatigue cycles at different stress levels, and tabulating creep life consumption by identifying time spent at high temperature and high stresses. In the GUI, these results will be used to highlight specific time frames in time series plots where operation led to high creep and fatigue life consumption and highlight locations where damage risk was high on a three-dimensional representation of the superheater. In addition to assessing creep and fatigue life, the software is expected to also estimate oxide growth and perform failure rate analyses of header stubs.

Before any analysis is performed, the software should help the user identify what data is needed to perform each analysis. It should provide a simple interface that allows the user to select what data they have available and display what analyses the tool can perform with that data. Alternatively, the interface should allow the user to select a specific analysis that they are interested in and display the specific data that is required to perform that analysis. The user should be able to launch a new analysis from this interface.

Once a dataset is chosen, the software should identify each column in the data file and allow users to define various settings and thresholds defining the pre-processing checks for their superheater system. Once an analysis is chosen and data is uploaded, the software should pre-process the data to rectify errors, such as rate of change checks and stuck value checks, and ensure consistency with previously existing datasets through statistical analyses. These analyses will verify that the distributions of new data align with historical data, ensuring reliable comparisons and assessments. Upon uploading time series data, users should also be able to input geometric information and material properties essential for the selected analyses.

The software's GUI should enable users to map all uploaded sensors onto a three-dimensional representation of the superheater system, enhancing the visualization of analysis results. For basic risk assessment analyses, the software should be able to aggregate load, temperature, flow rate, and pressure data over fixed time intervals (e.g., hourly). This aggregated data will be processed using a cycle counting algorithm to identify recurring fatigue cycles. It will evaluate periods of high-temperature operation under significant stress to assess high-risk periods concerning creep life. The results will be tabulated for the user, and the superheater representation will be color-coded to highlight regions exposed to the highest risk.

For advanced creep and fatigue assessments, users will be able to provide detailed geometry and material properties of the superheater. This additional information will allow the software to calculate the specific stresses encountered, resulting in a more accurate assessment of consumed creep and fatigue life. Similar to the basic assessments, the software is designed to tabulate the magnitude and frequency of fatigue cycles and the duration spent at high temperatures under substantial stresses. It will display a color-coded graphical representation of the superheater to facilitate user comprehension of the results.

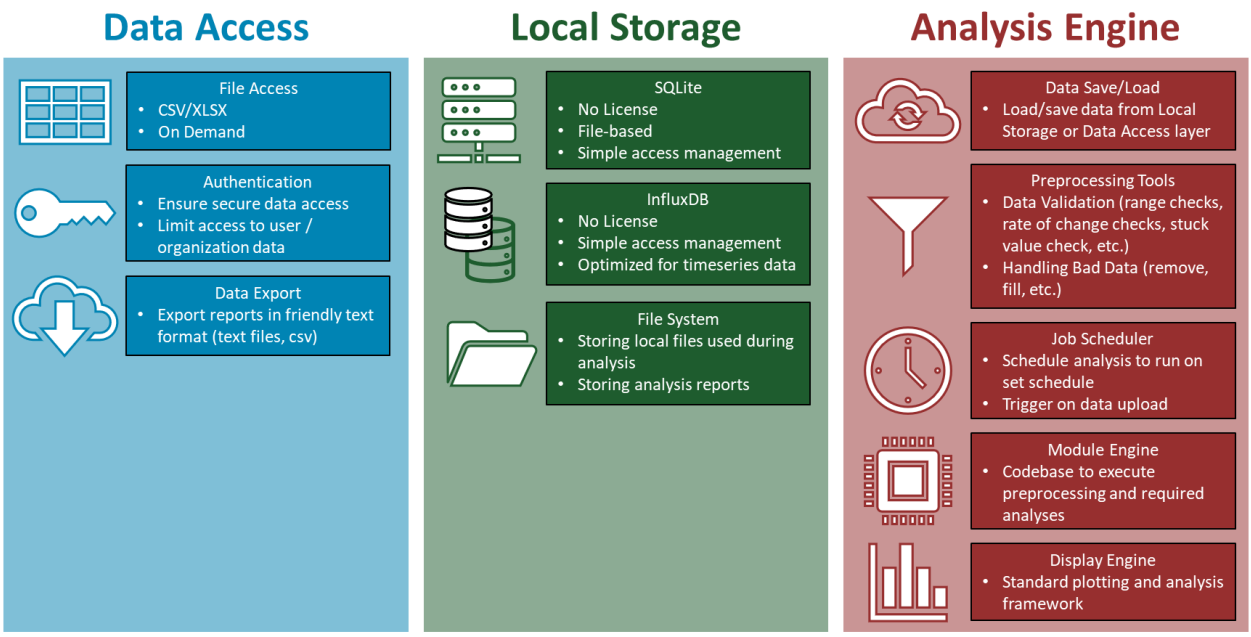
In addition to assessing creep and fatigue life, the software is designed to estimate oxide growth over time using the supplied time series data, leveraging an existing EPRI oxide thickness calculator. The user should also be able to upload data regarding header stub failure rates, along with the previously discussed time series data, to perform a Weibull analysis to predict when the rate of stub failures will increase significantly. This analysis will consider both the operational time to failure and the operating conditions in the Weibull regression.

The software generates outputs that help users identify specific regions of their superheater headers and tubing experiencing increased damage based on operational history. It calculates relative damage observed for each mechanism analyzed and for each specific hardware component that has been mapped. These outputs enable users to understand the primary damage risks associated with their superheater operations and identify which headers or tubes are incurring the most damage. Users can incorporate these insights into their asset management systems to optimize maintenance planning for their superheater systems.

- **System Features**

The EPRI Superheater Damage Evaluation Software framework is intended to be developed as a web application designed to be hosted on EPRI servers. It will feature a web-based interface that allows users to upload data, define features for their coal boiler or heat recovery steam generator (HRSG) superheater system, and view analysis that informs the user of operation that has been particularly harmful to the superheater in terms of creep and fatigue life.

The software is designed to have three core functional areas: Data Access, Local Storage, and an Analysis Engine, each crucial for analyzing the operation of their superheater hardware.



Data Access

Purpose/Overview: Handling of dataset in either CSV or XLSX formats.
Main Requirements:

- Establish protocols for handling file-based data for on-demand processing.
- Convert and integrate varied data file types into the analytics workflow

Integration Points: Data files are received through the software's GUI and are passed to Data Pre-processing for standardization.

Data Requirements

Time Series Data

Purpose/Overview: Define the required format of the data files uploaded to the software.
Main Requirements:

- **Data Format and Structure:** The system should accept historian extracts as CSV files. Each CSV file is expected to have timestamps in ISO 8601 format, with 'Z' indicating UTC time in the first column. Header columns to the right of the timestamp column should include data fields including temperature, pressure, and flow rate for each location. Each data field (temperature, pressure, flow rate) is expected to specify its unit of measurement specified in the header row (e.g., Temperature (°C), Pressure (MPa), Flow Rate (Kg/s)). The software should validate units of measurement in the uploaded CSV files and flag any deviation with an error message.

- Example CSV Structure

```
Timestamp, Temp_C_Loc1A1, Press_MPa, Flow_Kg/s, Meas_Unit_Loc1,
2024-06-18T00:00:00Z, 350, 2.5, 1000, value1, ...
2024-06-18T00:01:00Z, 352, 2.6, 1010, value1, ...
2024-06-18T00:02:00Z, 353, 2.7, 1020, value1, ...
```

- **Data Retrieval and Access:** Users will manually retrieve and upload data from their historian system into the software. The software is designed to parse the CSV files and validate the format before importing them into the software's time series database. Users will export data from their historian system in CSV format, ensuring the file meets the specified structure and units of measure listed above. The CSV file can then be uploaded through the software's data upload interface. Upon uploading, the software should parse the CSV file and provide real-time feedback on the validation results. If the format is incorrect, specific error messages will guide the user on necessary corrections.
-
- **Data Content:** Uploaded data will include key parameters such as temperature, pressure, and flow rate for each specified location. Each parameter must be clearly labeled in the CSV header for accurate parsing and integration. The temperature should be in the range of 0°C to 1000°C, pressure from 0 to 10 MPa, and flow rate from 0 to 5000 Kg/s.
  - **Specific Naming Convention Required:**
    - Measurement type: [Temp, Press, Flow, Load]
    - Units of Measure: [Temp (C, R, F, K), Press (MPa, kPa, psia, psig), Flow (kgQs, lbmQs, kgQhr, lbmQhr), Load (MW)]
    - Location ID: Arbitrary alphanumeric ID (unique)
    - For each data column, the measurement type, units of measure, and location number are specified as follows:
      - MeasurementType_UnitsofMeasure_LocationID
        - e.g., Temp_R_Loc15A2,
        - Press_psia_A,
        - Press_psia_B
      - The software will be responsible for identifying and parsing out locations, types of measurements, and units of measure. Uploaded data should be imported into SQLite and Influx databases. The user will be responsible for mapping tag location IDs to a specific physical location in the software interface.

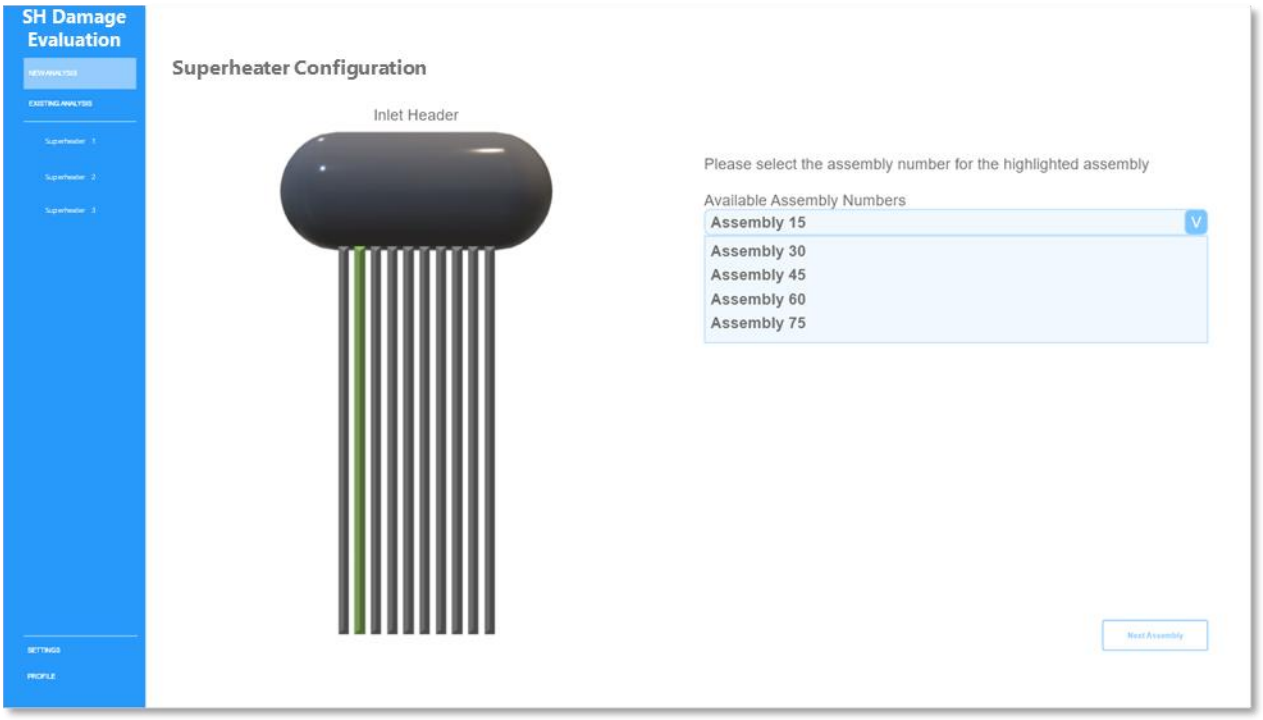
Location Data (Tube Numbering and Location)

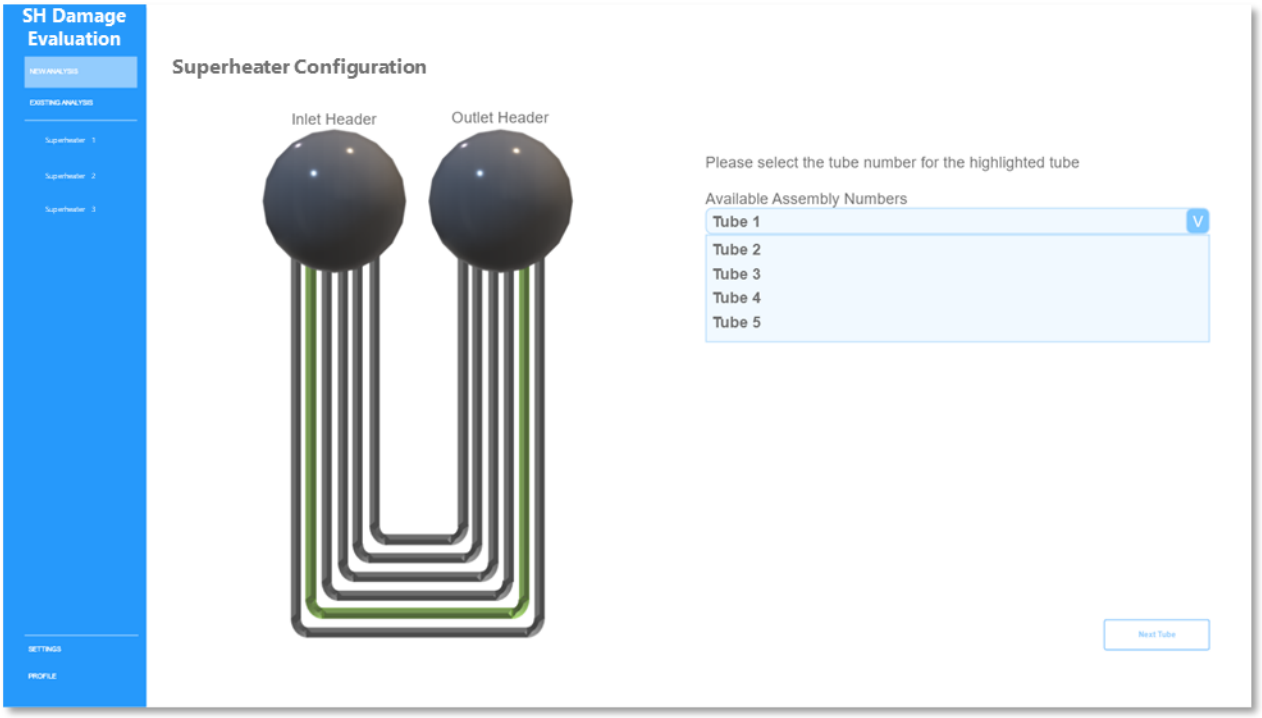
Purpose/Overview: Define the location data required to map uploaded sensor data to a digital representation of the superheater system.

Main Requirements:

- **Data Structure and Storage:** Mapping of time-series data should be entered through the user interface. The user interface should provide the ability to specify the number of:
  - Tube assemblies (wall to wall)
  - Rows per assemblies (number of "U"s)
  - Number and type of heat exchangers (e.g., SH or RH)

  The software assumes a "U" shaped tube path for each tube row. After specifying the number of tube bundles, the software is designed to create prepopulated representations and numbering. The software should walk the user through mapping each assembly by highlighting one assembly and prompting the user to choose an assembly number from a drop down. Once each assembly is mapped, the software should walk the user through assigning tube numbers using a similar interaction. All assembly and tube numbers should be parsed from the data column names, which will use the specified format above.





Geometry and Material Information

Purpose/Overview: Define the geometrical data that is required for the software to perform analyses of different superheater systems.

Main Requirements:

- **Data Content:** Geometry data should include:
  - Header inner and outer diameter, wall thickness, length
  - Tube inner and outer diameter, wall thickness
  - Tube length for each segment in the 'U' shape for each tube bundle and tube number
- Material properties data should include specifications such as material type and other relevant properties (e.g., tensile strength, thermal conductivity). Material properties should include tensile strength (MPa), thermal conductivity (W/mK), and other relevant factors like corrosion resistance. The user will be able to create a library of typical tube geometries (and material) which can be assigned to each tube or each bundle.
- **Data Management:** Users will provide geometry and material information in structured formats (xlsx/csv) that the software can upload through the user interface. This meta-data is stored in a centralized SQLite database, accessible for analysis and assessment. Users will have access to this data through a dedicated interface for querying and analysis.

Authentication

Purpose/Overview: Manages secure access to data by validating user credentials and imposing role-based permissions, thus safeguarding data integrity.
Main Requirements:

- Support for basic, Windows, OAuth, and token-based authentication methods.
- Enable role-based and claims-based access control, including impersonation features for specific data sources.

Security: Employ stringent practices for the storage of authentication data and secure token-based system access.
Integration Points: Controls are applied to data access methods, including Historian Access, File Access, and APIs.

Data Export

Purpose/Overview: Equips users with the ability to export reports, allowing for offline review of analyses and incorporation into external workflows.
Main Requirements:

- Intuitive configuration options for exporting data to standard data formats, including but not limited to flat files and databases.
- Maintain the integrity of data through all export processes and offer options for custom export formats.

Local Storage

Purpose/Overview: Provides versatile, robust data storage framework that effectively organizes different data types and manages efficient data retrievals.
Main Requirements:

- Supports varied storage mechanisms, including SQLite, InfluxDB, and the Windows file system to cater to data storage preferences.
- Facilitates data integrity and consistency with save/load operations incorporating meta-tagging for efficient storage utilization.
- Enable mapping between time-series data tags and standard measurement names to aid data organization and retrieval.

Integration Points: Direct interface with Data Pre-processing for receiving data, serving as the primary repository for the software's analysis.

Analysis Engine

Data Save/Load

Purpose/Overview: Facilitates a fluid transfer of data for processing within the Analysis Engine, bridging the gap between data storage and analysis.
Main Requirements: Efficiently load data from Local Storage or Data Access and save processed data back, ensuring data is correctly cached for quick access and reliability.

Preprocessing Tools

Purpose/Overview: A collection of data preparation tools designed for cleaning and transforming data, setting the stage for analysis and modeling.
Main Requirements: Offer a range of preprocessing capabilities, from cleaning datasets to advanced data transformation operations.

- **Range Checks:** Verify that data values fall within predefined acceptable ranges for each parameter (e.g., temperature, pressure, flow rate). Define acceptable value ranges for each parameter in the software configuration. During data ingestion, each data point will be checked against these ranges.
  - **Parameter Ranges:** Define the acceptable ranges for each parameter: Temperature (0°C to 1000°C), Pressure (0 to 10 MPa), Flow Rate (0 to 5000 Kg/s).
  - **Example Configuration:**

```
{
  "Temperature": {
    "min": 0,
    "max": 1000,
    "unit": "°C"
  },
  "Pressure": {
    "min": 0,
    "max": 10,
    "unit": "MPa"
  },
  "FlowRate": {
    "min": 0,
    "max": 5000,
    "unit": "Kg/s"
  }
}
```

- **Rate of Change Checks:** Ensure that the change in data values between consecutive time steps is within expected limits. Specify maximum allowable rate of change for each parameter. The software should compare the rate of change between consecutive data points and flag any values that exceed these limits.
  - **Rate of Change Limits:** Define maximum allowable rates of change for each parameter.
  - **Example Configuration:**

```
{
  "Temperature": {
```

```
    "maxRateOfChange": 5,
    "unit": "°C/min"
  },
  "Pressure": {
    "maxRateOfChange": 0.1,
    "unit": "MPa/min"
  },
  "FlowRate": {
    "maxRateOfChange": 100,
    "unit": "Kg/s"
  }
}
```

- **Stuck Value Checks:** Detect if a parameter value remains constant for an unusually long period, indicating a potential sensor malfunction. Set a threshold for the maximum number of consecutive identical values. The software will flag data points if a value remains unchanged beyond this threshold.
  - **Example Configuration:**

```
{
  "Temperature": {
    "maxConsecutiveIdenticalValues": 10
  },
  "Pressure": {
    "maxConsecutiveIdenticalValues": 10
  },
  "FlowRate": {
    "maxConsecutiveIdenticalValues": 10
  }
}
```

- **Units of Measure Checks:** Ensure that data values are reported in the correct units of measure as specified in the configuration. Define expected units of measure for each parameter. The software will validate that incoming data is reported in these units.
  - **Units Specification:** Clearly define the expected units for each parameter in the configuration file.
  - **Example Configuration:**

```
{
  "Temperature": {
    "unit": "°C"
  },
  "Pressure": {
    "unit": "MPa"
  },
  "FlowRate": {
    "unit": "Kg/s"
  }
}
```

- **Missing/Blank/Corrupt Value Checks:** Identify and handle missing, blank, or corrupt data values. Check for missing or blank values and verify data integrity. Any rows containing such values will be flagged and handled
  - **Example Configuration:**

```
{
```

```
    "Temperature": {
      "handleMissingValues": "flag"
    },
    "Pressure": {
      "handleMissingValues": "flag"
    },
    "FlowRate": {
      "handleMissingValues": "flag"
    }
  }
```

Handling Bad Data

Purpose/Overview: If data fails any validation check, the entire row for that channel and time step should be removed by the software, or the value will be skipped or imputed as best fits the type of analysis being performed. The software is not designed to perform interpolation or any modification of existing data. All corrections are left to the user.
Main Requirements:

- **User Instructions:** Provide clear instructions on how users should correct bad data outside the software.
- **Summary Report:** Provide users with a summary report of bad data for each channel.
- **Content:**
  - Total number of rows processed.
  - Number of rows removed per validation check (range, rate of change, stuck value, units, missing/blank/corrupt values).
  - Breakdown of bad data by channel (e.g., temperature, pressure, flow rate).
- **Format:** The report is generated in a structured format such as a CSV or PDF file for easy review and analysis.
- **Example Summary Report Format:**

```
Summary Report of Bad Data

Channel Total Rows Processed   Rows Removed (Range Check)     Rows      Removed
(Rate of Change)       Rows Removed (Stuck Value)     Rows    Removed    (Units)
        Rows Removed (Missing/Blank/Corrupt)  Total Rows Removed
Temperature    10000          50                             10
                      5                              0
        15                                    80
Pressure       10000          30                             20
                      0                              0
        10                                    60
Flow Rate      10000          25                             15
                      10                             5
        20                                    75
Total Rows Processed: 30000
Total Rows Removed: 215
```

- **Automatic Filtering:** The software will automatically filter out invalid data based on the validation checks. Users will be responsible for correcting any issues via a fresh data upload. No in-software correction mechanisms will be provided.

Job Scheduler

Purpose/Overview: Automates and manages task execution based on specific triggers or schedules to ensure timely processing within the system.
Main Requirements: Schedule tasks at specified intervals or times, and facilitate immediate processing when required.

Analysis Engine

Purpose/Overview: Executes multiple code (e.g., Python) modules or scripts for advanced data analysis as needed, offering an additional layer of flexibility and computational complexity.
Main Requirements: Seamlessly integrate code scripts implementing different analyses within the Analysis Engine while maintaining system integrity and performance.

Dynamic Analysis Selection

1. **Graphical User Interface (GUI) Guidance:** The software should include a GUI to help users understand the data requirements for each analysis and the analyses available based on their data.
2. **Analyses and Data Requirements:** The software should contain a dictionary linking analyses to the required data, geometry, and material properties. (Relationships between analysis and data requirements are found in Technology-specific Requirements).
    - **Required Fields:**
        - **Analyses:** [“startup assessment”, “shutdown assessment”, “load transient assessment”, “general assessment”, “fatigue analysis”, “creep analysis”, “stub failure analysis”, “oxidation”]
        - **Data:** [“temperature”, “pressure”, “flow”, “load”]
        - **Geometry:** [“header ID”, “header OD”, “header thickness”, “header length”, “tube ID”, “tube OD”, “tube thickness”, “tube length”, “assemblies”, “tubes per assembly”]
        - **Material Properties:** [“LMP”, “modulus of elasticity”, “thermal expansion coeff”, “Poisson’s ratio”]
    - **Example Dictionary Structure:**

```
{
   "coal boiler": {
      "startup assessment": {
         "components": ["tubes"],
         "data":  ["temperature",  "pressure",  "flow",
         "load"],
         "geometry": ["assemblies", "tubes per assembly"]
      },
      "shutdown assessment": {
         "components": ["tubes",
         "data":  ["temperature",  "pressure",  "flow",
         "load"],
```

```
            "geometry": ["assemblies", "tubes per assembly"]
        },
    },
    "hsrg": {
        "startup assessment": {
            "components": ["header", "tubes"],
            "data": ["temperature", "pressure", "flow",
            "load"],
            "geometry": ["assemblies", "tubes per assembly"]
        },
        "shutdown assessment": {
            "components": ["header", "tubes",
            "data": ["temperature", "pressure", "flow",
            "load"],
            "geometry": ["assemblies", "tubes per assembly"]
        }
    }
}
```

3. **Interactive Data and Analysis Identification Tools:** The software is designed to provide tools for users to interact with the data requirements and analysis capabilities dynamically.
    - **Analysis Identification Tool:** Allows users to select available data and presents a list of analyses possible with the selected data. This will be accomplished using several panes, as defined below:
        - **System Toggle:** Shows a list of all boiler types considered (i.e., coal boilers and HRSGs). Users can select one system.
        - **Data Pane:** Displays all time-series data types defined in the analysis dictionary. Users can make multiple selections.
        - **Geometry Pane:** Displays all types of geometry defined in the analysis dictionary. Users can make multiple selections.
        - **Material Properties Pane:** Displays all material properties defined in the analysis dictionary. Users can make multiple selections.
        - **Analysis Pane:** For display purposes only. Based on user selections in other panes, it filters and displays possible analyses.
    - **Data Identification Tool:** This tool will allow the user to choose the type of system and the desired analysis and present the user with a list of data, geometry, and material properties required. This will be accomplished using several panes, as defined below:
        - **System Toggle**: Shows a list of all boiler types considered (i.e., coal boilers and HRSGs). Users can select one system.
        - **Analysis Pane:** Shows a list of all analyses defined in the analysis dictionary. Users can make one selection at a time.
        - **Data Pane:** For display purposes only. Shows time series data required for the chosen analysis based on system and analysis selections.
        - **Geometry Pane:** For display purposes only. Shows geometric information required for the chosen analysis.

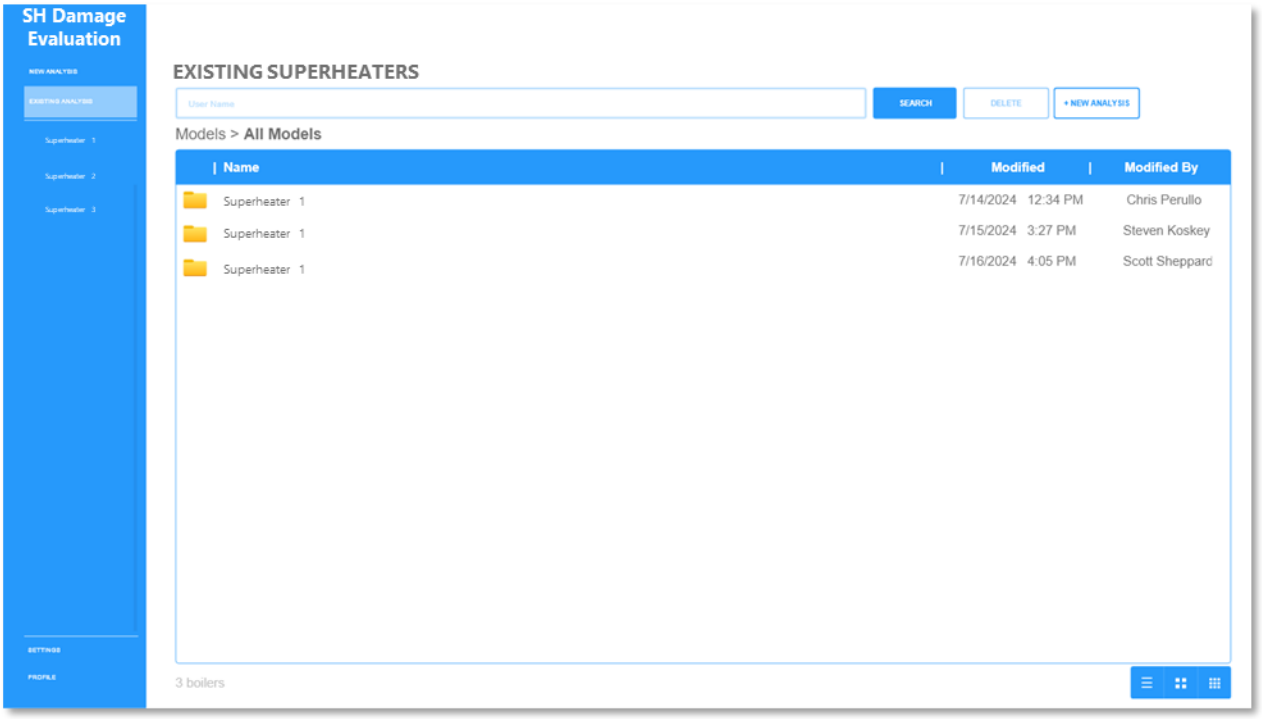
- **Material Properties Pane**: For display purposes only. Shows material properties required for the chosen analysis.

Display Engine

Purpose/Overview: Delivers a comprehensive suite of visualization tools, making data interpretation and analysis more accessible through standardized plotting and analysis modules.
Main Requirements: Compatibility with various data structures from the Analysis Engine, enhanced by user-friendly visual representation options.

- **User Interface**

The User Interface (UI) is designed to be straightforward, minimizing distractions and maximizing task efficiency. The dedicated sidebar should provide clear navigation between different parts of the application, enhancing focus and quick access to functionalities.
The application's first main screen (example seen below) should be compose of a simplified, user-friendly layout.



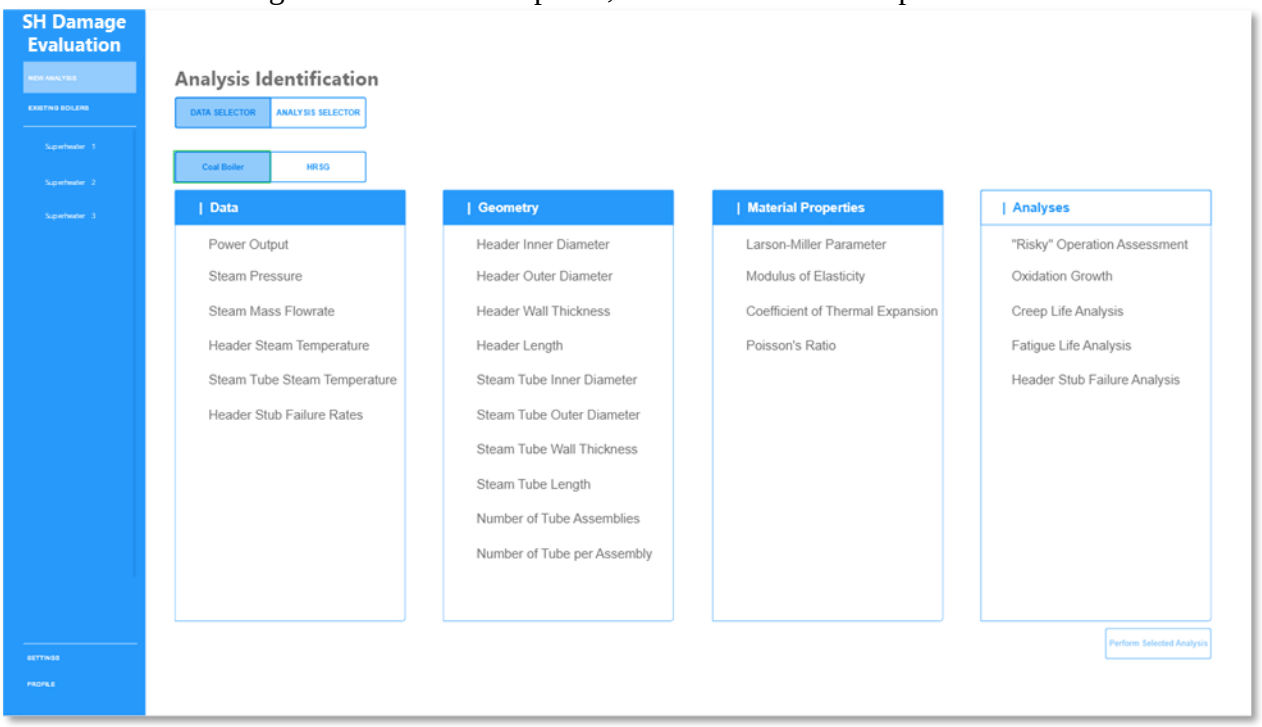
The sidebar, prominently displayed, should include the application's logo at the top and key navigation options like “New Analyses”, “Existing Analyses”, and a list of recently performed analyses, with each leading the user to the appropriate pages.
The main content area should comprise of a larger pane for detailed information regarding models and their categories, allowing users to create new and view old analyses.

Analysis Database

The software should include a GUI to help users understand the required data for each analysis and the analyses available based on the data type selected. Clear, interactive guidance on data and analysis requirements will help mitigate data limitations and operational constraints.

Analysis Identification Tool

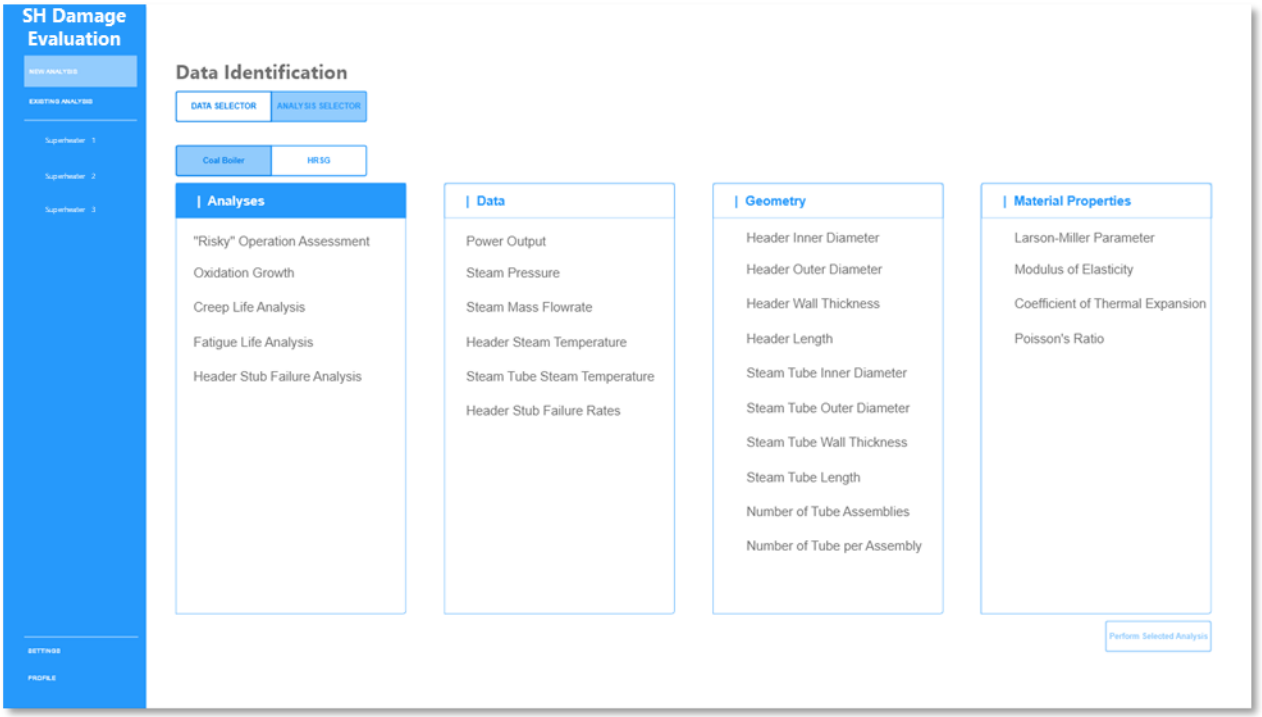
This tool shall allow users to select available data and will present a list of feasible analyses. This will be achieved using several interactive panes, as shown in the example below.



- System Selection Toggle: Lists all boiler types considered by the system (i.e., coal boilers and HRSGs) and allows selection.
- Data Pane: Displays all time series data types that are defined in the analysis dictionary, allowing multiple selections.
- Geometry Pane: Displays all geometry types defined in the analysis dictionary, allowing multiple selections.
- Material Properties Pane: Displays all material properties that are defined in the analysis dictionary, allowing multiple selections.
- Analysis Pane: This pane is for display purposes only. Based on the information provided by the user in the other panes, it will filter the list of analyses accordingly, and display only those analyses that are possible with the information provided by the user.

Data Identification Tool

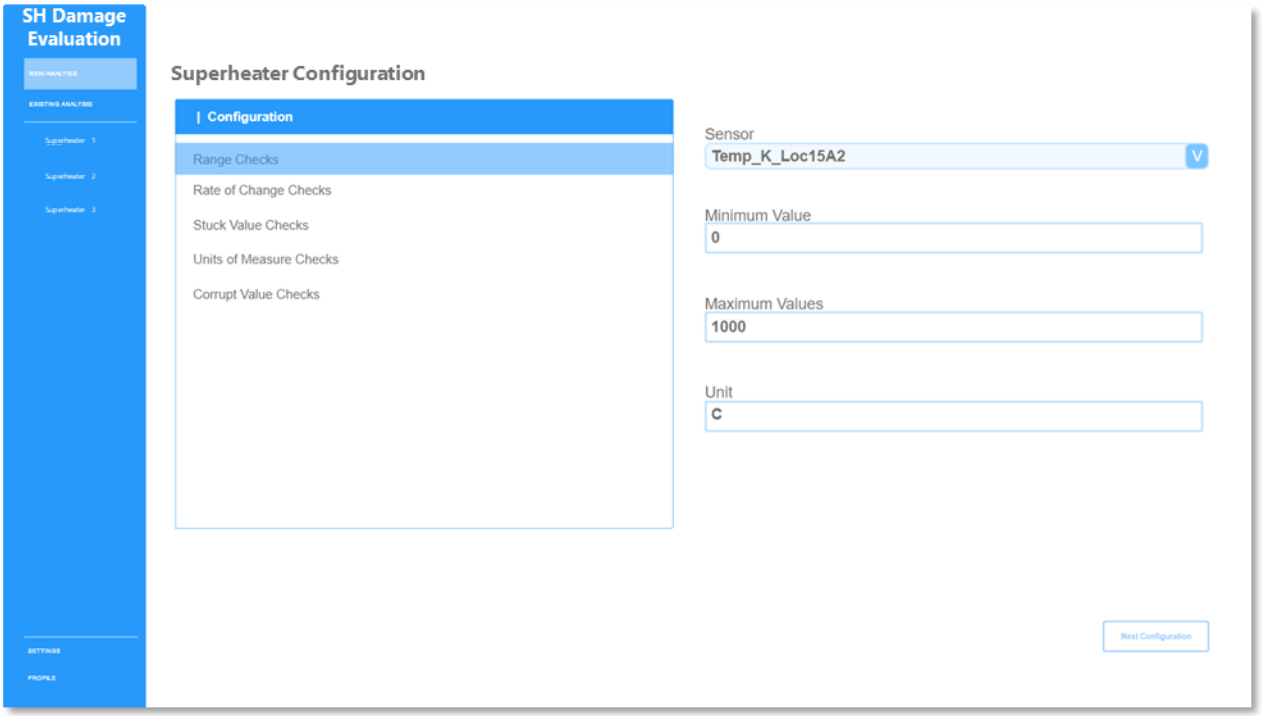
This tool will allow the user to choose the type of system and the desired analysis and should then present the user with required data, geometry, and material properties.



- System Selection Toggle: Lists of all boiler types considered by the system (i.e., coal boilers and HRSGs) and allows selection.
- Analysis Pane: Displays all analyses defined in the analysis dictionary and allows selection.
- Data Pane: This pane is for display only. Based on the information provided in the Analysis Pane, it will filter the data required accordingly, and display only the data required for the chosen analysis.
- Geometry Pane: This pane is for display only. Based on the information provided in the Analysis Pane, it will filter the geometric information accordingly, and display only the information required for the chosen analysis.
- Material Properties Pane: This pane is for display only. Based on the information provided in the Analysis Pane, it will filter the material properties accordingly, and display only the information required for the chosen analysis.

New Analysis Screens

After selecting an analysis from the analysis selection database, the software should prompt the user to choose a file for analysis. The file should meet the naming requirements described in Data Requirements.
Once the file is selected, the software is designed to guide the user through specifying each of the pre-processing configurations (as defined in Preprocessing Tools) step-by-step. The GUI should clearly indicate which set of thresholds is being defined at any given time and present the user with text fields to define each one. Although the process is linear, users should be able to move back-and-forth between the different configurations as needed. Below is an example of a configuration screen.



After defining the thresholds, the software should apply the pre-processing transformations to the dataset and upload the data to internal storage. A pre-processing report should be generated (as defined in
Handling Bad **Data**) and made available to the user through the software GUI.

Location Mapping Screen

While pre-processing and storing the data, the software should prompt the user to define the superheater for analysis. It will start by requesting basic information about the superheater to map data to specific hardware components, including the number of assemblies and how many tubes there are per assembly. The software is designed to generate a visual representation of the superheater hardware based on the information provided. Users should then be able to map the locations, as defined in Location Data (Tube Numbering and Location), to the visual representation of the superheater.

Geometry Information Input Screen

The software should provide a basic form for users to input geometric information about the superheater system. Required information includes:

- For Analyzing Headers
    - Header Inner Diameter
    - Header Outer Diameter
    - Header Wall Thickness
    - Header Length
- For Analyzing Steam Tubes
    - Steam Tube Inner Diameter
    - Steam Tube Outer Diameter
    - Steam Tube Wall Thickness
    - Steam Tube Length (for each of the tubes in the assembly)

Users will be able to specify whether the measurements are in metric or English units, and the software will convert units as needed for analysis.

Material Properties Input Screen

The software should provide a form for entering material properties that is similar to the geometric information form. Required information includes:

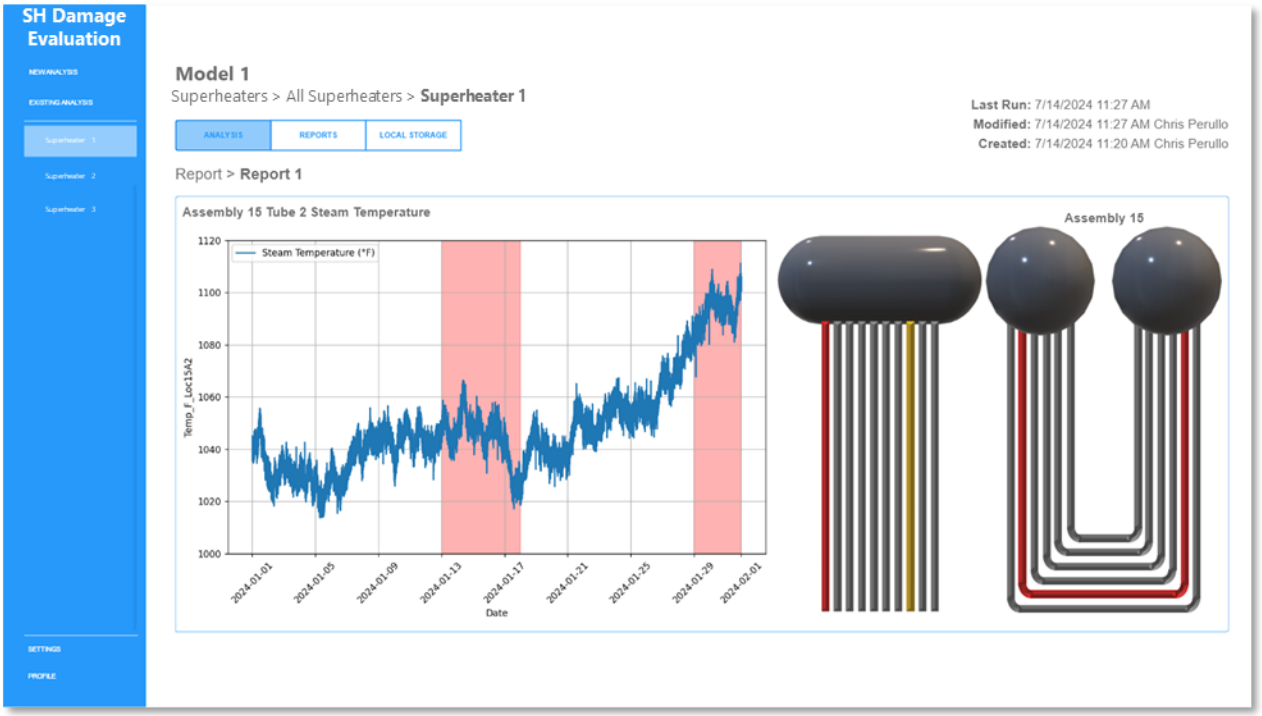
- Larson Miller Parameter
- Modulus of Elasticity
- Coefficient of Thermal Expansion
- Poisson's Ratio

The user will be able to specify whether the information has been entered in English or metric units, and the software should convert, as need, when performing analyses.

Superheater Analysis Screens

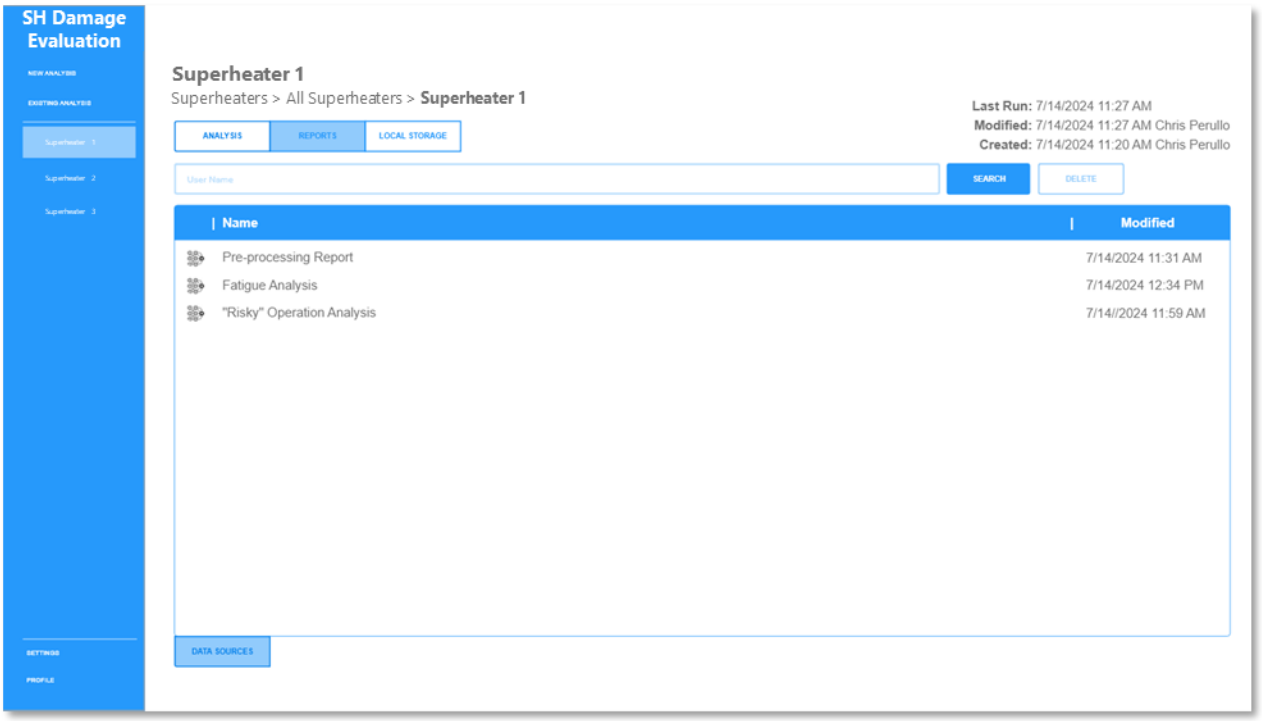
Analysis

When configuring a new system or revisiting the software, the software should default to showing the results of the most recent analyses, including pre-processing steps applied to the uploaded data. If multiple analyses exist for a given superheater system, it will benchmark the latest analysis against previous ones to alert the user if their system is performing as expected or if operational risk is increasing.
The software is designed to display analysis results graphically for easy digestion. The primary goal is to provide insight into specific locations where operational behavior has led to high creep and fatigue damage in the superheater system. Header and steam tube data will be mapped to a 3-dimensional representation of the superheater system to aid the user in understanding where specifically in their superheater system that operational risk observed has been highest. An example of this display is provided below.


SH Damage Evaluation
NEW ANALYSIS
EXISTING ANALYSIS
Superheater 1
Superheater 2
Superheater 3
SETTINGS
PROFILE
Model 1
Superheaters > All Superheaters > Superheater 1
ANALYSIS
REPORTS
LOCAL STORAGE
Last Run: 7/14/2024 11:27 AM
Modified: 7/14/2024 11:27 AM Chris Perullo
Created: 7/14/2024 11:20 AM Chris Perullo
Report > Report 1
Assembly 15 Tube 2 Steam Temperature
Steam Temperature (°F)
Temp_F_Loc15A2
1120
1100
1080
1060
1040
1020
1000
2024-01-01
2024-01-05
2024-01-09
2024-01-13
2024-01-17
2024-01-21
2024-01-25
2024-01-29
2024-02-01
Date
Assembly 15

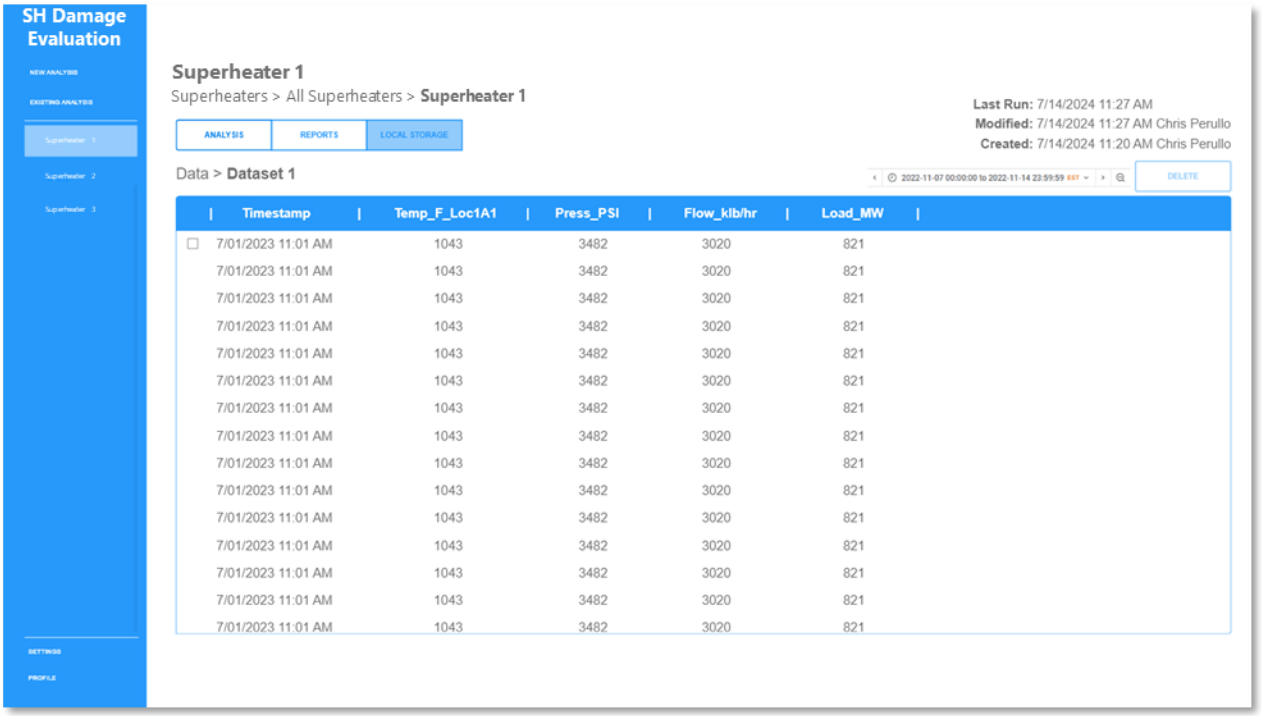
Reports

In the section titled “Reports”, the software should list all available reports, organized by superheater system. This will include data pre-processing reports and any performed analyses. Each report should clearly identify the type of analysis and the dataset used. Selecting a report will redirect the user to the Analysis screen, displaying the results of that analysis. An example display is below.



Local Storage

The Local Storage section of the software should contain information for the user to review datasets that have been uploaded to the software. It should provide summary metadata to identify what the dataset contains, when it was uploaded, who uploaded it, and what analyses it has been used in. If a dataset is selected, it will also provide the user with a raw data view to verify what data is contained within the dataset. A date and time filter should also be available to narrow the view of data to a specific time window of interest. Finally, records will be able to be selected individually or as a group and manually deleted by the user. A sample screen displaying a view of a data upload is shown below.


SH Damage Evaluation
NEW ANALYSIS
EXISTING ANALYSIS
Superheater 1
Superheater 2
Superheater 3
SETTINGS
PROFILE
Superheater 1
Superheaters > All Superheaters > Superheater 1
ANALYSIS
REPORTS
LOCAL STORAGE
Last Run: 7/14/2024 11:27 AM
Modified: 7/14/2024 11:27 AM Chris Perullo
Created: 7/14/2024 11:20 AM Chris Perullo
Data > Dataset 1
2022-11-07 00:00:00 to 2022-11-14 23:59:59 EST
DELETE
Timestamp | Temp_F_Loc1A1 | Press_PSI | Flow_klb/hr | Load_MW
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821
7/01/2023 11:01 AM 1043 3482 3020 821

➢ Non-Functional Requirements

The non-functional requirements for the Superheater Damage Evaluation Software outline the expected quality standards. The software must not only function correctly, but also handle user data securely and scale to accommodate multiple users. These requirements and the reporting functionalities are critical to the software's success and user adoption.

- Performance Metrics

Scalability and Response Time

Purpose/Overview: Ensure that the software can accommodate large data volumes and multiple users while still providing timely analysis results.
Main Requirements:

Efficient Algorithms: Implement algorithms optimized to minimize memory usage and processing time. Use suitable data structures and techniques, such as vectorization, to reduce operations.

Parallel Processing: Utilize multi-threading and multi-processing to handle multiple users and improve response times. An appropriate number of users to support should be determined at the start of software development, with the capability to adjust this number based on actual usage patterns and performance testing.

Data Management: Implement data compression and indexing to reduce the memory footprint and enhance access speeds. Store data in a well-indexed database for quick retrieval.

Data Processing Time Requirements: Analyses performed by the software shall be completed such that a user can submit an analysis and receive results in a timely manner that does not discourage further usage of the tool. An appropriate benchmark should be set at the start of software development.

Status Feedback: The software's GUI should provide real-time updates on data uploads and processing, including progress, estimated time remaining, and clear feedback to enhance the user experience and manage expectations.

Benchmarking: Conduct regular performance benchmarking tests during development to evaluate performance and research usage. Use benchmark results to identify optimization areas.

Analysis Results Reporting

Purpose/Overview: Generate clear, understandable reports for each analysis. Provide tabulated results for download and graphical displays in the GUI, linking risk assessments to a 3D representation of the superheater.
Main Requirements:

Analysis Reports: Summarize results for each of the following analyses.
Startup Assessment: Provide summaries for cold, warm, hot, and fast startups detailing changes in temperature, pressure, and flow during these events, establishing baseline changes and identifying startups significantly faster than the baseline.
Shutdown Assessment: Summarize changes in temperature, pressure, and flow during shutdowns, identifying shutdowns that deviate significantly from the baseline.
Transient Assessment: Summarize rapid changes in temperature, pressure, flow, and load data, highlighting any notable deviations.
General Assessment: Detail the frequency and duration of load cycling, time spent at high temperatures, and load cycles occurring at high temperatures, providing an overview of the operational stresses.
Fatigue Cycle Analysis: Summarize the number of stress cycles encountered by the superheater at various stress magnitudes, offering insights into the wear and tear experienced over time.
Creep Life Analysis: Provide a summary of the time spent at elevated temperatures and stress magnitudes, which are critical factors in assessing the remaining life and risk of creep damage.
Oxidation Analysis: Estimate oxide thickness growth over time.
Header Stub Failure Rate Analysis: Summarize header stub failure rates, including predictions for when failure counts are expected to rise rapidly, based on historical data.

- **Security Standards**

Authentication

Purpose/Overview: Manage secure access to data by appropriate parties by validating user credentials and imposing role-based permissions.
Main Requirements:

Role-Based Access Control (RBAC): Restrict data access based on user roles. Clearly define roles and their permissions to ensure minimal necessary access. Conduct regular audits of role assignments to prevent privilege creep. Apply the principle of least privilege to reduce impact of security breaches. Separate critical tasks among different roles to enhance security and operational integrity. Establish role hierarchies to manage permissions and access control efficiently.
Deployment Recommendation:
Automated Provisioning: Implement automated provisioning systems for managing user access including approval workflows. Conduct regular access reviews and enforce immediate de-provisioning when users leave or change roles.
Secure Storage of User Credentials: Manage and store user credentials securely using hashing and encryption. Restrict access to manage user credentials to authorized

personnel only.

Activity Logging: Log all access and activities related to sensitive data, including user identity, accessed resources, actions taken, and timestamps. Ensure logs are secure and tamper-evident. Use masked or tokenized representations to protect user privacy.

Data Transmission

Purpose/Overview: Ensure secure data transmission from the user's local system to the Software's local storage.

Main Requirements:

HTTPS: Use HTTPS for all web communications to ensure secure data transmission. Use certificates from trusted Certificate Authorities (CAs) and ensure proper installation.

Data Storage

Purpose/Overview: Ensure that data is securely stored and only accessible to those with the appropriate permissions.

Main Requirements:

InfluxDB Encryption: Data stored in InfluxDB should be secured through comprehensive encryption measures. This includes disk encryption, either for the entire disk or specific partitions where InfluxDB data is stored, to protect data at rest. The performance impact of disk encryption on InfluxDB operations should be evaluated, and encryption keys must be securely managed and rotated regularly. Additionally, transport encryption should be enforced using TLS to secure data in transit to and from InfluxDB. Proper TLS configuration, including the use of strong cipher suites and secure versions, is essential. TLS certificates should be regularly updated and managed to ensure ongoing security.

SQLite Database Access Control: Access to SQLite databases should be restricted using Role-Based Access Control (RBAC). Roles and their permissions must be clearly defined, with regular audits conducted to prevent unauthorized access. Specific roles include a Global Administrator, likely an EPRI employee, who manages the software and has full access for troubleshooting and maintenance; a Group Administrator, typically one per EPRI member, who can access all data and analyses within their organization; and individual Users, who can access data they have uploaded and analyses they have performed. Strong authentication methods and strict password policies, including regular changes and the use of MFA, should be enforced for database users.

File Access Control: Set strict file permissions to control access, adhering to the principle of least privilege. Audit logs must be maintained for file access and modifications, ensuring accountability. Use secure methods, such as encrypted email or secure file transfer protocols, for sharing CSV files, with access tracking to monitor who accessed and shared these files.

InfluxDB Access Control: Use strong authentication methods, including strict password policies, for InfluxDB users. Fine-grained permissions are implemented to control access to specific databases and measurements.
Deployment Recommendations:

SQLite Database Encryption: Use Transparent Data Encryption (TDE) to encrypt the entire database at the file level, with secure key management practices to avoid expiration and vulnerabilities. Column-level encryption may be applied to sensitive data, with careful consideration of performance impacts and access controls to ensure only authorized users can access encrypted data.

CSV File Encryption: Encrypt CSV files using strong algorithms, such as AES-256, with secure key management and regular rotation. Use reliable encryption tools and store encrypted files in secure locations, with strict access controls limiting decryption capabilities.

Spreadsheet Encryption: Utilize built-in encryption features of spreadsheet software, such as password protection in Excel, ensuring strong password management. Verify compatibility with users' software versions and platforms. External encryption tools, like GPG or AES, may also be used, with secure key management and data integrity checks, such as checksums, to verify the integrity of encrypted files.

Audit: Regular audits of database and file storage read/write permissions and user activities are essential to ensure security and compliance.

Data Integrity Checks: Implement data integrity checks, conduct regular audits to detect and correct data integrity issues, and use automated processes for verifying file integrity.

Regular Backups: Schedule regular backups of files and databases and store them securely. Employ backup encryption for additional security, and test backup restoration procedures regularly to ensure reliability. Redundant storage across multiple locations is recommended to prevent data loss, and a disaster recovery plan should be implemented and tested.

Secure Storage: Securely configure databases, with unnecessary services disabled and secure defaults used. Store files in secure, access-controlled locations, with regular security updates and patches applied. Monitor access to storage locations for suspicious activities.

Data Masking: Use dynamic data masking for non-production environments, with masked data employed in testing and development settings. Implement consistent data masking policies across all storage methods to protect sensitive information.

Comprehensive Logging: Enable detailed logging of database activities and file access and modifications, with measures in place to ensure logs are tamper-evident and secure. Do not log sensitive data directly and conduct regular log reviews to identify suspicious activities.

Real-time Monitoring: Real-time monitoring of database activities and file access and modifications is crucial, with alerts set up for suspicious activities. Develop and maintainan incident response plan to address potential security incidents promptly and effectively

➢ Analysis Requirements

The analysis requirements for the Superheater Damage Evaluation Software detail the system's analytical capabilities. These requirements specify the types of boilers to analyze, the types of analyses to perform, and the required data and meta-data necessary for these analyses.

- **Technology-specific Requirements**

Purpose/Overview: Support analysis for both coal boiler and HRSG systems. These are similar technologies with comparable measurements (albeit different designs and geometry). Analyses to be performed for each system are defined.

Main Requirements:

"Risky" Operation Analysis: The software is designed to analyze data on temperature, load, pressure, and flow rate to summarize stress cycles and the time spent at extreme temperatures. It should identify startup, shutdown, normal load fluctuations, and extreme ramp rates from load data. The software will also assess steam temperatures at the header and tubes for extreme values and fluctuations during startup, shutdown, and load changes. Additionally, it should quantify the number, magnitude, and frequency of cycles between different states of load, temperature, pressure, and flow rate.

Cycle Counting: Use a cycle counting algorithm, such as the range-pair or rainflow method, to identify different stress cycles in the data.

Fatigue Cycle Analysis: Adding geometric data and material properties enables the software to calculate hardware stresses to assess fatigue damage. Apply cycle counting algorithms to these calculated stresses to tabulate the number of cycles with different stress magnitude changes.

Creep Life Analysis: Adding geometric data and material properties enables the software to calculate the time spent at high temperatures under high stress conditions to assess creep damage. The software should summarize this time to provide a comprehensive view of creep life consumed.

Oxidation Analysis: Integrate an existing EPRI calculator, either through direct integration or by retooling it, to estimate oxide growth. This estimation will give users a benchmark to compare their own measurements against.

Header Stub Failure Rate Analysis: Given data on header stub failures, the time of failure, and other related data channels, the software should be able to perform a Weibull analysis to estimate stub failure rates over the superheater's lifespan.

Coal Boiler Analyses & Data Required

Table D. 1. Data Required for Coal Boiler Analyses

Analysis	Method	Header	Steam Tubes	Load	Flow Rate	Steam Pressure	Steam Temp(s)	Stub Failure Rates
"Risky" Operation	Assess freq. & severity of temp. changes during startup		Y	Y	Y	Y	Y	
	Assess freq. & severity of temp. changes during shutdown		Y	Y	Y	Y	Y	
	Assess freq. & severity of temp. changes during load changes		Y	Y	Y	Y	Y	
	Assess freq. & severity of temp. changes during load generating operation		Y	Y	Y	Y	Y	
Oxidation	Correlation		Y				Y	
Creep Life Consumed	Tabulating time duration of temp-stress loading at various condition	Y	Y	Y	Y	Y	Y	
Fatigue Life Consumed	Tabulating cycles at different stress condition		Y	Y	Y	Y	Y	
Header Stub Failure Rate Analysis	Weibull Analysis	Y						Y

Table D. 2. Geometry Required for Coal Boiler Analyses

Analysis	Method	Header ID/OD	Header Thickness	Header Length	Steam Tube ID/OD	Steam Tube Thickness	Steam Tube Length	# Tube Assemblies	Tubes per Assembly
"Risky" Operation	Assess freq. & severity of temp. changes during startup								
	Assess freq. & severity of temp. changes during shutdown								
	Assess freq. & severity of temp. changes during load changes								
	Assess freq. & severity of temp. changes during load generating operation								
Oxidation	Correlation								
Creep Life Consumed	Tabulating time duration of temp-stress loading at various condition	Y	Y	Y	Y	Y	Y	Y	Y
Fatigue Life Consumed	Tabulating cycles at different stress condition				Y	Y	Y	Y	Y
Header Stub Failure Rate Analysis	Weibull Analysis								

Table D. 3. Material Properties Required for Coal Boiler Analyses

Analysis	Method	LMP	Modulus of Elasticity	Coefficient of Thermal Expansion	Poisson's Ratio
"Risky" Operation	Assess freq. & severity of temp. changes during startup				
	Assess freq. & severity of temp. changes during shutdown				
	Assess freq. & severity of temp. changes during load changes				
	Assess freq. & severity of temp. changes during load generating operation				
Oxidation	Correlation				
Creep Life Consumed	Tabulating time duration of temp-stress loading at various condition	Y	Y	Y	Y
Fatigue Life Consumed	Tabulating cycles at different stress condition		Y	Y	Y
Header Stub Failure Rate Analysis	Weibull Analysis				

HRSG Analyses & Data Required

Table D. 4. Data required for HRSG Analyses

Analysis	Method	Header	Steam Tubes	Load	Flow Rate	Steam Pressure	Steam Temp(s)	Stub Failure Rates
"Risky" Operation	Assess freq. & severity of temp. changes during startup	Y	Y	Y	Y	Y	Y	
	Assess freq. & severity of temp. changes during shutdown	Y	Y	Y	Y	Y	Y	
	Assess freq. & severity of temp. changes during load changes	Y	Y	Y	Y	Y	Y	
	Assess freq. & severity of temp. changes during load generating operation	Y	Y	Y	Y	Y	Y	
Oxidation	Correlation		Y				Y	
Creep Life Consumed	Tabulating time duration of temp-stress loading at various condition	Y	Y	Y	Y	Y	Y	
Fatigue Life Consumed	Tabulating cycles at different stress condition		Y	Y	Y	Y	Y	
Header Stub Failure Rate Analysis	Weibull Analysis	Y						Y

Table D. 5. Geometry required for HRSG Analyses

Analysis	Method	Header ID/OD	Header Thickness	Header Length	Steam Tube ID/OD	Steam Tube Thickness	Steam Tube Length	# of Tube Assemblies	Tube per Assembly
"Risky" Operation	Assess freq. & severity of temp. changes during startup								
	Assess freq. & severity of temp. changes during shutdown								
	Assess freq. & severity of temp. changes during load changes								
	Assess freq. & severity of temp. changes during load generating operation								
Oxidation	Correlation								
Creep Life Consumed	Tabulating time duration of temp-stress loading at various condition	Y	Y	Y	Y	Y	Y	Y	Y
Fatigue Life Consumed	Tabulating cycles at different stress condition				Y	Y	Y	Y	Y
Header Stub Failure Rate Analysis	Weibull Analysis								

Table D. 6. Material Properties required for HRSG Analyses

Analysis	Method	LMP	Modulus of Elasticity	Coefficient of Thermal Expansion	Poisson's Ratio
"Risky" Operation	Assess freq. & severity of temp. changes during startup				
	Assess freq. & severity of temp. changes during shutdown				
	Assess freq. & severity of temp. changes during load changes				
	Assess freq. & severity of temp. changes during load generating operation				
Oxidation	Correlation				
Creep Life Consumed	Tabulating time duration of temp-stress loading at various condition	Y	Y	Y	Y
Fatigue Life Consumed	Tabulating cycles at different stress condition		Y	Y	Y
Header Stub Failure Rate Analysis	Weibull Analysis				

- **Data Collection Constraints**

Purpose/Overview: Define the minimum data required by the software to perform analyses of superheater systems.
Main Requirements:

Time Series Data: At a minimum, steam pressure, steam mass flow rate, plant load, and at least one steam temperature will be required. This data set allows for a summary of the "risky" operations experienced by the superheater. Additionally, the number of tube assemblies and steam tubes per assembly are necessary to scale the steam mass flow rate to an individual steam tube, summarizing the "risky" operations experienced by each tube. With additional temperature sensors, the analysis can extend to other regions of the superheter system.
Geometric Data: Incorporating geometric data allows the software to calculate system stresses. For headers, the required data includes inner and outer diameters, wall thickness, and length. For steam tubes, the necessary data includes inner and outer diameters, wall thickness, length, the number of assemblies, and the number of tubes per assembly.
Material Properties: Adding material properties, in addition to geometric, extends stress analyses to include relative creep and fatigue life consumption. For fatigue life analysis, the modulus of elasticity, coefficient of thermal expansion, and Poisson's ratio are required. Additionally, the Larson-Miller parameter is needed for creep life analysis.
Header Stub Failure Rates: At a minimum, age at failure and/or time since last maintenance are required for a basic Weibull analysis of stub failure rates. Additional parameters, such as operating temperature and pressure, can also be included to refine failure rates based on the operational profile, not just operational time.

- Software Maintenance

Purpose/Overview: The software should have ample resources (further defined in Performance Metrics) to support their operational history and years of remaining life. Specific data requirements for each analysis are defined in Technology-specific Requirements.
Main Requirements:

Deployment Period: The software is designed to provide a framework in which additional analyses are added over time. The data storage, handling, and visualization framework are independent of the analyses, such that new analyses can be added to the software as resources allow, providing the opportunity for a long operation life of the software.
Regular Updates: Software updates will introduce new features, address minor bugs, enhance performance, and ensure compliance with evolving industry standards. These updates will help maintain the software's relevance and efficiency over time.
User Feedback: The software should include an integrated feedback form that allows users to send feedback via email to an EPRI-hosted account. This feedback will be aggregated and considered for future updates. Additionally, EPRI will host regular meetings with members to discuss services, including the software. During these meetings, members are able to provide feedback, which may be incorporated into future software updates.

➢ User Requirements

The user requirements for the Superheater Damage Evaluation Software specify the need for flexible data access and configurable pre-processing thresholds. These features will ensure the software adapts to various superheater systems.

- **User Profiling and Customization**
-

Purpose/Overview: Allow the user to configure certain thresholds defining how their specific system is analyzed, enabling the software to remain flexible to differences in the operation of coal boilers and HRSGs.

Main Requirements:

Configuration Files: Accept various configuration files (examples in Preprocessing Tools) that define data cleaning and analysis processes tailored to specific systems. These configurations will ensure the software remains adaptable to different boiler systems, regardless of operating temperatures and pressures. High-level definitions of these configurations are provided below:

- **Range Checks:** Allows users to set minimum and maximum values for each incoming data channel to ensure data integrity.
- **Rate of Change Checks:** Enables users to define acceptable rates of change for each data channel over specified time periods, helping to detect anomalies.
- **Stuck Values:** Permits users to specify the interval length that a data channel must report a constant value before it is flagged as an issue.
- **Units of Measure:** Users can specify the unit of measure for each data channel, ensuring consistency in data interpretation.
- **Handling of Errors:** Provides options for users to decide how data errors are managed, including flagging for review, automatic filling (e.g., forward fill, backward fill, interpolation), or removal.

User-Defined Settings: Users will define these configurations through a graphical user interface (GUI), providing flexibility for any boiler system. The GUI will guide the user step by step through each configuration, clearly stating the configuration being defined and providing text fields for the required values. After completing each configuration, the user can save it, and the GUI will proceed to the next one. For each data channel, the following values are required:

- **Range Checks:** Minimum Value, Maximum Value, Units of Measure (e.g., °C, MPa, etc.)
- **Rate of Change Checks:** Maximum Rate of Change, Units of Measure (e.g., °C/min, MPa/min, etc.)
- **Stuck Values:** Maximum Consecutive Identical Values
- **Units of Measure:** Unit (e.g., °C, MPa, etc.)
- **Handling of Errors:** Handling Missing Values (e.g., “flag”, “remove”)

Once a configuration is created, users should be able to assign it an alphanumeric name that represents the component being analyzed. The software should also allow users to import existing configurations for additional sensors. For instance, a configuration defined for a steam tube temperature sensor can be applied to any other steam tube temperature sensors as needed.

➢ System Requirements

The Superheater Damage Evaluation Software must securely store data, scale with increasing data uploads, and manage aging datasets. These requirements ensure the system can quickly recall recent data and efficiently archive older, less frequently accessed data.

- **Data Management**

Scalability Requirements

Purpose/Overview: Ensure that the software scales appropriately under high data load and user count.
Main Requirements:

Handling Varying Data and Users: Each user should have a separate environment and database to ensure data isolation and privacy. The time series data is stored in an InfluxDB database, while metadata is managed in a set of SQLite databases.

Expected Data Volume: The system should support up to 5 million rows and 1000 channels per site, allowing for up to 10 years of data retention at one-minute intervals per channel.

Performance Benchmarks: Consistent performance is crucial, regardless of data volume. Benchmarks must include data retrieval times, data processing speeds, and system response times under varying loads.

Strategies for Ensuring Performance: Although load balancing is not currently anticipated to be necessary, efficient data indexing techniques in both InfluxDB and SQLite should be employed to expedite data retrieval and queries. Additionally, caching strategies should be implemented for frequently accessed data to reduce the load on the database.

Handling Peak Loads: While scaling is not currently anticipated to be required, the system design should remain flexible to accommodate future scalability if necessary.

Data Storage Requirements

Purpose/Overview: Ensure that sufficient data storage is available to accommodate many users of the software uploading years' worth of data for their superheater systems.
Main Requirements:

Capacity and Duration: Utilize InfluxDB to store large volumes of time series data, supporting up to 5 million rows and 1000 channels. SQLite should handle the storage of metadata and other non-time-series data, which, although smaller in volume, is equally important.

Archival Compression: Implement downsampling to compress older data, such as reducing the data older than one year to five-minute intervals. This process must ensure that the downsampled data retains an acceptable level of accuracy for subsequent analysis and assessment.

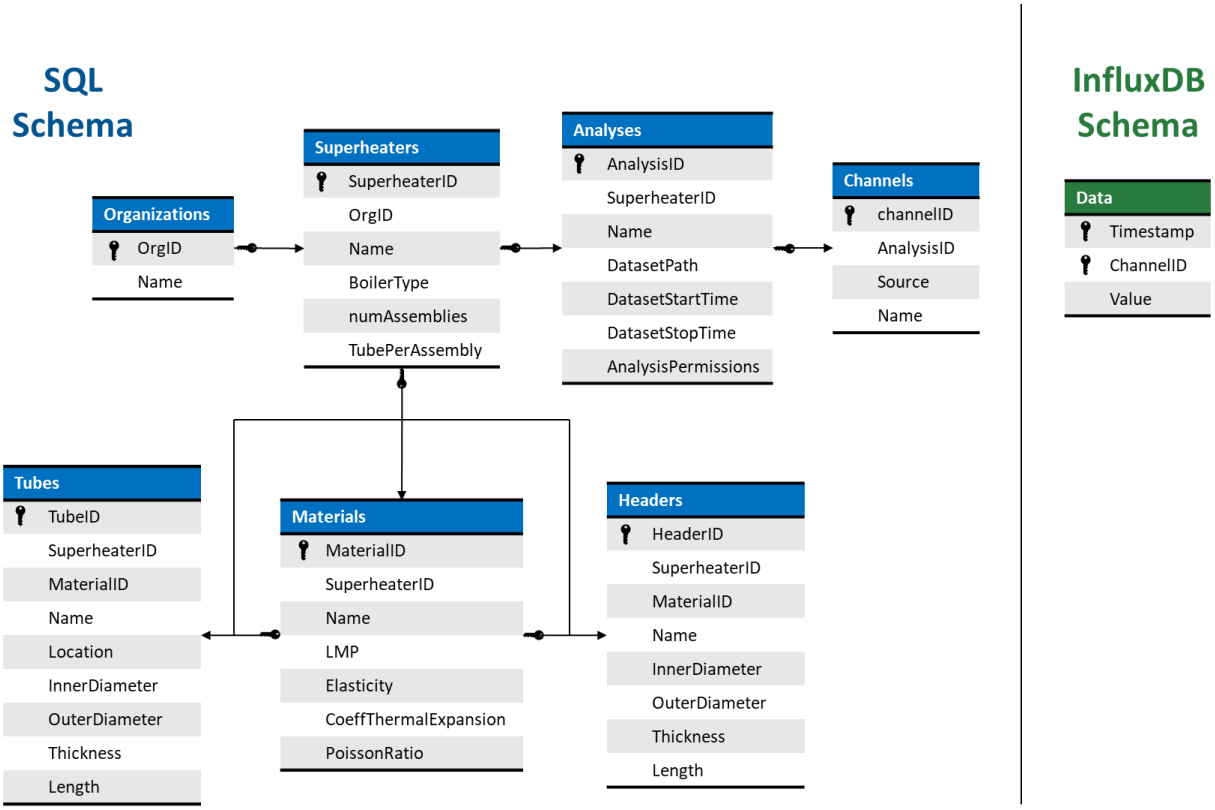
Data Maintenance: Conduct regular database cleaning to remove obsolete or redundant data. Employ archival strategies to store old data in a compressed format, preserving storage space while maintaining data accessibility. Additionally, implement automated purging policies to delete data that is no longer required, based on predefined retention periods.

Database Schema: All information about the superheater systems and analyses should be stored in SQLite databases, with the exception of the raw time series data which should be stored in an Influx database.

By using several SQLite tables to store metadata about the users and their superheater systems, overall storage requirements will be reduced. Foreign keys from linked tables can be used to establish relationships between users, their organization, and what superheaters and analysis results they have access to. Below is a figure showing the different databases that will be required and the foreign keys shared between them. The required SQLite databases are defined as follows:

- Organizations: Contains the name of each organization using the software and assigns them a unique organization ID.
- Superheaters: Defines names for each superheater, what organization they belong to, assigns a unique superheater ID, and stores basic information about boiler type and the number of assemblies and how many tubes are in each assembly.
- Analyses: Defines a name for each analysis, assigns a unique analysis ID to each analysis, identifies what superheater was analyzed, what dataset was used, start and end dates for the data used, and has a flag for whether or not EPRI can use this dataset for further research purposes.
- Channels: Defines the name of each channel, assigns a unique channel ID for each channel, and lists which analysis they are used in.
- Material Properties: Defines a name of the material, assigned a unique material ID for the material, identifies which superheater it belongs to, and contains a column for all expected material properties (defined in Technology-specific Requirements).
- Tubes: Defines a name for each tube, assigns a unique tube ID, identifies which superheater it belongs to and the appropriate material, identifies the location within the superheater, and contains the expected geometric information for that tube (defined in Technology-specific Requirements)
- Headers: Defines a name for each header, assigns a unique header ID, identifies which superheater it belongs to and the appropriate material, and contains the expected geometric information for that tube (defined in Technology-specific Requirements)

The time series data itself will be stored in an Influx database, named Data, with three columns: “Timestamp”, “ChannelID”, and “Value”. Information from queries in the SQLite tables pertaining to a specific analysis can be used to query the appropriate time series data from the Influx database. Database schemas are shown below.


SQL Schema
Organizations
OrgID
Name
Superheaters
SuperheaterID
OrgID
Name
BoilerType
numAssemblies
TubePerAssembly
Analyses
AnalysisID
SuperheaterID
Name
DatasetPath
DatasetStartTime
DatasetStopTime
AnalysisPermissions
Channels
channelID
AnalysisID
Source
Name
Tubes
TubeID
SuperheaterID
MaterialID
Name
Location
InnerDiameter
OuterDiameter
Thickness
Length
Materials
MaterialID
SuperheaterID
Name
LMP
Elasticity
CoeffThermalExpansion
PoissonRatio
Headers
HeaderID
SuperheaterID
MaterialID
Name
InnerDiameter
OuterDiameter
Thickness
Length
InfluxDB Schema
Data
Timestamp
ChannelID
Value

➢ Quality Attributes

The quality attributes of the Superheater Damage Evaluation Software detail the system's ability to ensure that incoming data is of high quality and complete with respect to what has been previous observed and that the analyses are sufficiently validated to ensure accurate analysis of superheater systems. These requirements ensure that the software provides valid and accurate analyses that provide meaningful results for users.

- **Quality of Data**

Maintaining Consistency of Data Over Time

Data Integration

Purpose/Overview: Define how new data for a previously analyzed superheater system will be incorporated to existing databases.
Main Requirements:

Frequency of Data Updates: Data from a single source may be periodically updated through new data uploads. Users should be able to append new data to the existing dataset during these uploads.
Data Appending Process: Ensure the format and structure of new datasets are consistent with any existing data. Perform data format checks, preventing duplication of records, and maintaining data integrity when appending new data.

Ensuring Data Consistency

Purpose/Overview: Provides a robust framework for ensuring that new file uploads contain data that reasonably matches the value and ranges of that which was uploaded previously.
Main Requirements:

Statistical Comparison: Perform a statistical comparison between new data and existing data to identify significant differences. This process should detect potential errors, such as discrepancies in units of measure or other anomalies. Use statistical methods like mean, median, and standard deviation to compare the new data with existing data for each parameter.
Data Validation During Appending: Check for unit consistency between new and existing data, compare the range of new data to the historical range to identify any outliers or unusual patterns, and assess the rate of change in the new data compared to the existing data to ensure it falls within expected limits, ensuring data consistency between datasets.
Handling Discrepancies: If statistically significant differences are identified, flag these discrepancies for user review and correction.
Example Configuration:

```
{
   "Temperature": {
      "mean": 150,
      "stdDev": 5,
      "unit": "°C"
   },
   "Pressure": {
```

```
            "mean": 10,
            "stdDev": 0.5,
            "unit": "MPa"
        },
        "FlowRate": {
            "mean": 500,
            "stdDev": 50,
            "unit": "Kg/s"
        }
    }
```

User Notification and Reporting

Purpose/Overview: Inform the user whether new data uploads appear to be consistent with data that has been uploaded for previous analyses on a given superheater system.
Main Requirements:

Notification of Inconsistencies: Notify users if any inconsistencies or significant statistical differences are detected during the data appending process.

Format: Notifications should be delivered through alerts within the software interface and summary reports.

Content of Notifications: Notifications should include a summary of statistically significant differences, a detailed list of the affected parameters and data points, and suggested actions for user review and potential corrective measures.

Notification Examples:

```
Notification Report of Data Inconsistencies

Parameter Existing Data Mean     New Data Mean  Difference     P-Value Status
Temperature     150°C                  155°C          5°C            0.01
  Significant Difference
Pressure 10 MPa                 9.8 MPa        0.2 MPa        0.05    Within
Range
Flow Rate 500 Kg/s              550 Kg/s       50 Kg/s        0.02
  Significant Difference

Summary of Discrepancies:
- Temperature: Statistically significant difference detected. Review unit
consistency and recent sensor calibration.
- Flow Rate: Statistically significant difference detected. Check for potential
errors in data logging or unit conversion.

Suggested Actions:
- Verify units of measure for new data uploads.
- Review recent calibration and maintenance records for affected sensors.
- Perform manual correction and re-upload the corrected data if necessary.
```

Ensuring Data Completeness

Purpose/Overview: Confirm that data uploads contain all of the necessary data prior to performing analyses, avoiding potential analysis errors as a result of missing data.
Main Requirements:

Data Completeness Checks: Verify the completeness of time series data against metadata for each tube, row, and bundle. During data upload, compare the time series data tags with the expected tags based on the metadata, ensuring all expected tags are present and have corresponding data points. The hierarchical structure of bundle, row, and tube identifiers will be used to verify the presence of data for each expected tag.

Handling Incomplete Data: Flag if any expected tags are missing or there are gaps in the time series data. Mark incomplete data rows and notify users for corrective action.

Example Configuration:

```
{
   "Bundle1_RowA_Tube1": {
                        "expectedTags": ["Temperature", "Pressure"],
                        "missingTagsAction": "flag"
        },
        "Bundle1_RowA_Tube2": {
                        "expectedTags": ["Temperature", "Pressure"],
                        "missingTagsAction": "flag"
        }
}
```

User Notification and Reporting: Immediately notify users if incomplete data is detected during the upload process. Generate alerts within the software interface and provide a detailed report for the user. Notifications will include a summary of the incomplete data detected, details of the missing tags and parameters, and suggested actions for user review and correction.

Notification Examples:

```
Notification Report of Incomplete Data

Bundle Number   Row Number     Tube Number    Expected Tags
Missing Tags           Status
Bundle1 RowA           Tube1          Temperature,          Pressure
Temperature            Incomplete
Bundle1 RowA           Tube2          Temperature,          Pressure
Pressure               Incomplete
Bundle2 RowB           Tube3          Temperature,     Flow     Rate
Temperature, Flow Rate Incomplete

Summary of Incomplete Data:
Bundle1_RowA_Tube1: Missing temperature data.
Bundle1_RowA_Tube2: Missing pressure data.
Bundle2_RowB_Tube3: Missing both temperature and flow rate data.

Suggested Actions:
Verify the completeness of data uploads before submission.
Ensure all sensors are operational and data logging is functioning
correctly.
Review and re-upload the complete data set for the affected tags.
```

- Quality of Analysis

Model Accuracy and Customization:

Development of Test Datasets

Purpose/Overview: Provide a robust set of test data that encompasses the full range of behavior of both coal boiler and HRSG systems, suitable for validating the analyses performed by the software.
Main Requirements:

Creation of Comprehensive Test Sets: Develop test datasets tailored for each specific analysis the software will perform, ensuring thorough testing during the development phase.
Known Outcomes: These test datasets should contain predefined outcomes to facilitate validation of the software's analytical capabilities. Tabulate number, magnitude, and frequency of cycles for each dataset prior to testing.
Scenario Coverage: Ensure the synthetic data covers a wide range of scenarios, including normal operation, startup and shutdown, and large and small transient events.
Historical Data: Historical datasets from EPRI members may be used to validate the analyses. The number of stress and thermal cycles is unlikely to be known a priori, and so will require manual labeling of the data. Historical data should cover the full range of operation of a typical superheater and include steam temperatures (both of the header and any steam tubes), steam pressure, steam flow rate, plant load, and header stub failure rates.
Synthetic Data: Synthetic datasets may be created for validation of the software. When creating the datasets, the developer will specify the number, magnitude, and frequency of stress and thermal cycles to occur in the datasets. This will provide exact targets for the software to identify. These synthetic datasets should cover the full range of steam temperature, pressure, flow rate, and plant load values typical of a superheater system.

Validation Processes

Purpose/Overview: Detailed validation reports should document the results, and any discrepancies, enhancing the models' reliability and accuracy.
Main Requirements:

Comparison with Known Outcomes: Validate the software's analytical models by comparing their analysis against the known outcomes of the test datasets.
Performance Metrics: Measure the software's ability to detect pre-determined events within the test datasets using metrics such as True Positives (TP) for correctly identified events, True Negatives (TN) for correctly identified non-events, False Positives (FP) for incorrectly identified events, and False Negatives (FN) for missed events. Additionally, for duration-based analyses (such as creep life consumed), accurately calculate the duration within a set tolerance. Appropriate benchmarks should be set for each analysis at the start of software development.
Documentation and Reporting: The software should be validated with both the initial release and subsequent updates, accompanied by detailed validation reports documenting the datasets used, results, and any discrepancies found. The software should be refined until all validation tests are passed, with all validation logs stored as records that all tests have been passed prior to each

software release.

Example Report:

Dataset	TP	TN	FP	FN	Result
Historical dataset 1	w	x	y	z	Pass
Synthetic dataset 1	a	b	c	d	Failed

User Communications:

The software should promptly notify users of any errors encountered during analysis, providing clear and detailed error messages along with potential causes and corrective actions. It should conduct pre-analysis data validation and continuous monitoring to identify anomalies, generating comprehensive error reports that summarize the issues and their impact on analysis results. Additionally, the software should maintain detailed error logs that can aid in future bug fixes.

User Notification of Errors

Purpose/Overview: Provide the user with actionable knowledge about any errors in the superheater analysis, such that they can adjust any inputs as needed.

Main Requirements:

Immediate Notification: Promptly notify the user if an error occurs during analysis, clearly indicating that the analysis was not completed. Notifications should be provided through in-app alerts.

Downloadable Reports: Users should also have the option to download a simple report detailed the errors encountered during the analysis

Potential Errors: Examples of potential errors include data format errors, incomplete or missing data, unexpected values or outliers, and calculation or analysis errors.

Detailed Error Descriptions: Provide detailed error messages that include the nature of the error and any relevant error codes to help users understand what went wrong.

Identification and Reporting of Data Issues

Purpose/Overview: Identify issues that arise either during data pre-processing or analysis, and inform them of corrective actions to improve data analysis.

Main Requirements:

Pre-Analysis Data Validation: Perform pre-analysis checks to identify any anomalies or errors in the data before starting the analysis. These checks should include range validation, consistency checks, and missing data detection. Results of data cleaning should be provided to the user.

Real-Time Error Detection: Continuously assess data during analysis to detect and report anomalies that may affect the accuracy of the results. For example, input data with a large amount of noise could lead to artificially high cycle counts.

User Guidance: Provide users with insights into the potential causes of the errors and suggest corrective actions, such as re-collecting data, correcting data entries, or checking sensor functionality.

Detailed Logs: Maintain detailed logs of all errors and anomalies encountered during data analysis. These logs should capture the time of occurrence, nature of the error, affected data, and

any corrective actions taken. General users should be able to access and review their own error logs to track recurring issues and identify patterns that may require attention, while developers should have access to all error logs to inform future software updates.

➢ Implementation Roadmap

The roadmap for implementing that Superheater Damage Evaluation Software is structured around a one-year development period, with six months of initial development, followed by three months of testing, and an additional three months of development to incorporate user and SQA feedback. Development should be conducted in two-week sprints, with feature prioritization occurring at the start of each sprint based on a combination of technical dependencies and risk management consideration. A development timeline is shown below.

	Task Name	Hours	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
Analysis Development	Data Pre-processing	120												
	Cycle Counting Algorithm	80												
	"Risky" Operation Analysis	160												
	Oxide Growth Estimation	80												
	Creep Assessment	160												
	Fatigue Assessment	160												
	Header Stub Failure Rate Analysis	160												
GUI Development	GUI Framework	80												
	Database Structure	80												
	Data Upload and Database Insertion	80												
	Develop location mapping display	160												
	Analysis and Data Identification Interfaces	120												
	Developing reporting displays	120												
Preproduction Testing	SQA Submission & Testing	16												
	Member Testing	40												
Revision & Resubmission	Software Revision	160												
	Final SQA Submission	16												
	Total Estimated Hours	**1792**												

Key Milestones:

- Months 1-6 (Development):
  - Sprint planning every two weeks.
  - Create and refine user stories and features; track and adjust development efforts as necessary.
  - Core analysis and software interface can be developed in parallel.
  - Proposed timeline:
    - Analysis Development:
      - Month 1: Begin implementing the required data cleaning checks and bad data handling processes.
      - Month 2: Implement the cycle counting algorithm. Finalize development of data cleaning and bad data handling processes.
      - Month 3: Being implementing the “Risky” Operation analyses.
      - Month 4: Finalize development of the “Risky” Operation analyses. Integrate oxide growth estimation calculation. Begin implementing creep life analysis.
      - Month 5: Finalize development of the creep life analysis functionality. Begin implementing both the fatigue life and header stub failure rate analyses.

        - Month 6: Finalize development of core algorithms. Finalize development of the fatigue life and header stub failure rate analyses.
    - GUI Development:
        - Month 1: Develop graphical interface framework and begin building framework of databases.
        - Month 2: Finalize database structures and begin development of tools for data uploading and insertion into the appropriate databases.
        - Month 3: Finalize implementation of data upload and insertion and begin development of location mapping display and analysis and data Identification tools.
        - Month 4: Finalize development of location mapping display and analysis and data identification tools.
        - Months 5-6: Begin developing appropriate displays for the different analyses as their respective analysis codes are complete.
- Months 7-9 (Preproduction Testing by EPRI SQA and Users):
    - Months 7-8 (SQA Testing)
        - Virus scanning.
        - Installation and compatibility testing.
        - Verification of software functionality against the user manual.
        - User manual tutorials validation.
    - Months 8-9 (Preproduction Testing by Users):
        - Test users will download, install, and utilize the software in their local environment.
        - Establish a straightforward feedback submissions process.
- Months 10-12 (Revision and Final Production Submission)
    - A single revision cycle based on SQA results and user feedback.
    - Implement changes and prepare deliverables for final SQA submission, including:
        - Required SQA submission forms.
        - Updated software specification.
        - Software installation files.
        - Source code.
        - User manual with tutorials.
- Post-Approval:
    - Upon EPRI SQA approval, release the software for member use on EPRI's website.

This roadmap is designed to be simple, allowing for flexibility for adjustments as needed throughout the process. Some aspects of the software, namely the analysis code and the graphical interface, can be developed in parallel to ensure that overall progress is never blocked by any one issue. All relevant documentation and deliverables required for each phase of the SQA process have been specified, and a post-approval period has been included to guarantee a smooth transition into the release and onboarding of EPRI members.

References

[1] EIA, International Energy Outlook 2023, 2023. https://www.eia.gov/outlooks/ieo/.

[2] V. Mladenov, V. Chobanov, A. Georgiev, Impact of Renewable Energy Sources on Power System Flexibility Requirements, Energies 14 (2021) 2813. https://doi.org/10.3390/en14102813.

[3] J. Rúa, A. Verheyleweghen, J. Jäschke, L.O. Nord, Optimal scheduling of flexible thermal power plants with lifetime enhancement under uncertainty, Applied Thermal Engineering 191 (2021) 116794. https://doi.org/10.1016/j.applthermaleng.2021.116794.

[4] J. Cochran, M. Miller, O. Zinaman, M. Milligan, D. Arent, B. Palmintier, M. O'Malley, S. Mueller, E. Lannoye, A. Tuohy, B. Kujala, M. Sommer, H. Holttinen, J. Kiviluoma, S.K. Soonee, Flexibility in 21st Century Power Systems, 2014. https://doi.org/10.2172/1130630.

[5] P. Sarda, E. Hedrick, K. Reynolds, D. Bhattacharyya, S.E. Zitney, B. Omell, Development of a Dynamic Model and Control System for Load-Following Studies of Supercritical Pulverized Coal Power Plants, Processes 6 (2018) 226. https://doi.org/10.3390/pr6110226.

[6] G. Liu, X. Yang, X. Yang, K. Liang, D. An, D. Wu, X. Ren, Typical Damage Prediction and Reliability Analysis of Superheater Tubes in Power Station Boilers Based on Multisource Data Analysis, Energies 15 (2022) 1005. https://doi.org/10.3390/en15031005.

[7] N. Kumar, P. Besuner, S. Lefton, D. Agan, D. Hilleman, Power Plant Cycling Costs, 2012. https://doi.org/10.2172/1046269.

[8] P. Myles, S. Herron, Impact of Load Following on Power Plant Cost and Performance : Literature Review and Industry Interviews, 2012. https://www.netl.doe.gov/projects/files/ImpactofLoadFollowingonPowerPlantCostandPerformance_100115.pdf.

[9] C. De Michelis, C. Rinaldi, C. Sampietri, R. Vario, Condition monitoring and assessment of power plant components, in: Power Plant Life Management and Performance Improvement, Elsevier, 2011: pp. 38–109. https://doi.org/10.1533/9780857093806.1.38.

[10] C. Dutta, J. Kumar, T.K. Das, S.P. Sagar, Recent Advancements in the Development of Sensors for the Structural Health Monitoring (SHM) at High-Temperature Environment: A Review, IEEE Sensors J. 21 (2021) 15904–15916. https://doi.org/10.1109/JSEN.2021.3075535.

[11] R. Lyon, The development of condition monitoring tools for the power generation industry, Insight - Non-Destructive Testing and Condition Monitoring 46 (2004) 516–519. https://doi.org/10.1784/insi.46.9.516.40848.

[12] V. Saini, D. Bhattacharyya, Nonlinear State and Parameter Estimation for Differential-Algebraic Equation Systems with Uncertain Algebraic Equations and Unknown Noise Covariance, in: AIChE, 2020. https://aiche.confex.com/aiche/2020/prelim.cgi/Paper/610813.

[13] J. Zhao, M. Netto, Z. Huang, S.S. Yu, A. Gomez-Exposito, S. Wang, I. Kamwa, S. Akhlaghi, L. Mili, V. Terzija, A.P.S. Meliopoulos, B. Pal, A.K. Singh, A. Abur, T. Bi, A. Rouhani, Roles of Dynamic State Estimation in Power System Modeling, Monitoring and Operation, IEEE Trans. Power Syst. 36 (2021) 2462–2472. https://doi.org/10.1109/TPWRS.2020.3028047.

[14] V. Saini, K. Reynolds, D. Bhattacharyya, Condition Monitoring of Non-Linear Differential-Algebraic Equation Systems with Colored Process and Measurement Noise, in: 2021 AIChE Annual Meeting, AIChE, 2021. https://aiche.confex.com/aiche/2021/meetingapp.cgi/Paper/631588.

[15] J.-X. Wang, Z. Wu, M.-L. Zhong, S. Yao, Data-driven modeling of a forced convection system for super-real-time transient thermal performance prediction, International Communications in Heat and Mass Transfer 126 (2021) 105387. https://doi.org/10.1016/j.icheatmasstransfer.2021.105387.

[16] J.D. Bendtsen, O. Sorensen, Simulation, state estimation and control of nonlinear superheater attemperator using neural networks, in: Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334), IEEE, Chicago, IL, USA, 2000: pp. 1430–1434 vol.2. https://doi.org/10.1109/ACC.2000.876737.
[17] J. Zhang, G. Hou, J. Zhang, Adaptive Neuro-control System for Superheated Steam Temperature of Power Plant over Wide Range Operation, in: Sixth International Conference on Intelligent Systems Design and Applications, IEEE, Jian, China, 2006: pp. 138–141. https://doi.org/10.1109/ISDA.2006.85.
[18] Liangyu Ma, Yongjun Lin, K.Y. Lee, Superheater steam temperature control for a 300MW boiler unit with Inverse Dynamic Process Models, in: IEEE PES General Meeting, IEEE, Minneapolis, MN, 2010: pp. 1–6. https://doi.org/10.1109/PES.2010.5589600.
[19] L. Ma, K.Y. Lee, Neural network based superheater steam temperature control for a large-scale supercritical boiler unit, in: 2011 IEEE Power and Energy Society General Meeting, IEEE, San Diego, CA, 2011: pp. 1–8. https://doi.org/10.1109/PES.2011.6039231.
[20] Z. Cui, J. Xu, W. Liu, G. Zhao, S. Ma, Data-driven modeling-based digital twin of supercritical coal-fired boiler for metal temperature anomaly detection, Energy 278 (2023) 127959. https://doi.org/10.1016/j.energy.2023.127959.
[21] J. Krzywanski, Heat Transfer Performance in a Superheater of an Industrial CFBC Using Fuzzy Logic-Based Methods, Entropy 21 (2019) 919. https://doi.org/10.3390/e21100919.
[22] X. Wu, J. Shen, Y. Li, K.Y. Lee, Fuzzy Modeling and Predictive Control of Power plant Steam Temperature System, IFAC-PapersOnLine 48 (2015) 397–402. https://doi.org/10.1016/j.ifacol.2015.12.411.
[23] I. Benyó, J. Kovács, J. Mononen, U. Kortela, Modelling of steam temperature dynamics of a superheater, International Journal of Simulation: Systems, Science and Technology 6 (2005). https://ijssst.info/Vol-06/No-6/Benyo.pdf.
[24] P. Wang, F. Si, Y. Cao, Z. Shao, S. Ren, Prediction of superheated steam temperature for thermal power plants using a novel integrated method based on the hybrid model and attention mechanism, Applied Thermal Engineering 203 (2022) 117899. https://doi.org/10.1016/j.applthermaleng.2021.117899.
[25] S. Prasanth, S. Narayanan, N. Sivakumaran, R.S. Sharanya Rominus, A hybrid technique to control superheater steam temperature in power plants using multi modeling and predictive sliding mode control, Sustainable Energy Technologies and Assessments 55 (2023) 102912. https://doi.org/10.1016/j.seta.2022.102912.
[26] D. Beahr, V. Saini, E. Hedrick, S.M.C. Hong, D. Bhattacharyya, Estimation-Based Model Predictive Control with State-Dependent Objective Prioritization; An Application to a Natural Gas Combined Cycle Power Plant, in: 2022 AIChE Annual Meeting, AIChE, 2022. https://aiche.confex.com/aiche/2022/meetingapp.cgi/Paper/648442.
[27] Y. Sato, M. Nomura, H. Matsumoto, M. Iioka, Steam Temperature Prediction Control for Thermal Power Plant, IEEE Trans. on Power Apparatus and Syst. PAS-103 (1984) 2382–2387. https://doi.org/10.1109/TPAS.1984.318376.
[28] J. Wallace, R. Clarke, The application of Kalman filtering estimation techniques in power station control systems, IEEE Trans. Automat. Contr. 28 (1983) 416–427. https://doi.org/10.1109/TAC.1983.1103238.
[29] K.L. Lo, Z.M. Song, E. Marchand, A. Pinkerton, Development of a static-state estimator for a power station boiler Part I. Mathematical model, Electric Power Systems Research 18 (1990) 175–189. https://doi.org/10.1016/0378-7796(90)90051-4.

[30] M. Basu, MODELLING APPROACH TO SUPERHEATER OF A BOILER, International Journal of Applied Engineering and Technology 2 (2012) 1–8.
[31] C.S. Somayajula, D. Bhattacharyya, X. Liu, Optimal Sensor Network Design and Nonlinear State Estimation for in-Situ Corrosion Monitoring in Coal-Fired Boilers, in: 2020 Virtual AIChE Annual Meeting, AIChE, 2020. https://aiche.confex.com/aiche/2020/meetingapp.cgi/Paper/610824.
[32] E. Boje, E. Eitelberg, Augmented kalman filtering for a superheated steam header system, IEEE Trans. Contr. Syst. Technol. 11 (2003) 773–781. https://doi.org/10.1109/TCST.2003.816416.
[33] Z. Patel, E. Boie, Application of the Continuous-Discrete Kalman Filter to State Estimation of the Convection Pass of a Once-Through Power Boiler, in: 2019 American Control Conference (ACC), IEEE, Philadelphia, PA, USA, 2019: pp. 3167–3173. https://doi.org/10.23919/ACC.2019.8815021.
[34] K. Węglarz, D. Taler, J. Taler, M. Marcinkowski, General numerical method for hydraulic and thermal modelling of the steam superheaters, Energy 291 (2024) 130371. https://doi.org/10.1016/j.energy.2024.130371.
[35] K.L. Lo, Y. Rathamarit, State estimation of a boiler model using the unscented Kalman filter, IET Gener. Transm. Distrib. 2 (2008) 917. https://doi.org/10.1049/iet-gtd:20080217.
[36] P. Nevriva, S. Ozana, M. Pies, L. Vilimec, Dynamical model of a power plant superheater, WSEAS Transactions on Systems 9 (2010) 774–783.
[37] S. Vijayalakshmi, D. Manamalli, S. Ramesh Babu, Modeling and Simulation of Industrial Coal Fired Boiler, AMM 787 (2015) 255–259. https://doi.org/10.4028/www.scientific.net/AMM.787.255.
[38] C.S. Somayajula, D. Bhattacharyya, X. Liu, S. Hu, Optimal Sensor Network Design for Corrosion Monitoring of Multi-Scale Time-Varying Systems: Application to a Power Plant Boiler, in: 2021 AIChE Annual Meeting, AIChE, 2021. https://aiche.confex.com/aiche/2021/meetingapp.cgi/Paper/631781.
[39] G. Dieck-Assad, J. Luis Vega-Fonseca, I. Hernández- Ramírez, A. Favela-Contreras, Modeling, Simulation, and Control of Steam Generation Processes, in: D. Cvetković (Ed.), Modeling and Computer Simulation, IntechOpen, 2019. https://doi.org/10.5772/intechopen.79410.
[40] W. Zima, S. Grądziel, A. Cebula, M. Rerak, E. Kozak-Jagieła, M. Pilarczyk, Mathematical model of a power boiler operation under rapid thermal load changes, Energy 263 (2023) 125836. https://doi.org/10.1016/j.energy.2022.125836.
[41] W. Zima, Numerical modeling of dynamics of steam superheaters, Energy 26 (2001) 1175–1184. https://doi.org/10.1016/S0360-5442(01)00078-0.
[42] W. Zima, Mathematical model of transient processes in steam superheaters, Forsch Ingenieurwes 68 (2003) 51–59. https://doi.org/10.1007/s10010-003-0106-7.
[43] C.S. Somayajula, D. Bhattacharyya, X. Liu, S. Hu, Optimal Sensor Network Design and State Estimation of Nonlinear Differential Algebraic System: Application to Corrosion Monitoring in a Power Plant Boiler, in: 2022 AIChE Annual Meeting, AIChE, 2022. https://aiche.confex.com/aiche/2022/prelim.cgi/Paper/648233.
[44] V. Saini, Hybrid Grey-Box Modeling and Estimation for Condition Monitoring of Complex Dynamics Energy Systems, in: 2023 AIChE Annual Meeting, AIChE, 2023. https://aiche.confex.com/aiche/2023/meetingapp.cgi/Paper/674637.
[45] J. Zhao, J. Qi, Z. Huang, A.P.S. Meliopoulos, A. Gomez-Exposito, M. Netto, L. Mili, A. Abur, V. Terzija, I. Kamwa, B. Pal, A.K. Singh, Power System Dynamic State Estimation: Motivations, Definitions, Methodologies, and Future Work, IEEE Trans. Power Syst. 34 (2019) 3188–3198. https://doi.org/10.1109/TPWRS.2019.2894769.

[46] K.L. Lo, Z.M. Song, E. Marchand, A. Pinkerton, Development of a static-state estimator for a power station boiler Part II. Estimation algorithm and bad data processing, Electric Power Systems Research 18 (1990) 191–203. https://doi.org/10.1016/0378-7796(90)90052-5.
[47] J. Opalka, L. Hubka, Nonlinear State and Unmeasured Disturbance Estimation for Use in Power Plant Superheaters Control, Procedia Engineering 100 (2015) 1539–1546. https://doi.org/10.1016/j.proeng.2015.01.526.
[48] A.T. Nair, T.K. Radhakrishnan, K. Srinivasan, S. Rominus Valsalam, Kalman Filter Based State Estimation of a Thermal Power Plant, in: 2011 International Conference on Process Automation, Control and Computing, IEEE, Coimbatore, Tamilnadu, India, 2011: pp. 1–5. https://doi.org/10.1109/PACC.2011.5978971.
[49] G. Jonsson, O.P. Palsson, K. Sejling, Modeling and Parameter Estimation of Heat Exchangers—A Statistical Approach, Journal of Dynamic Systems, Measurement, and Control 114 (1992) 673–679. https://doi.org/10.1115/1.2897740.
[50] G. Jonsson, O.P. Palsson, An Application of Extended Kalman Filtering to Heat Exchanger Models, Journal of Dynamic Systems, Measurement, and Control 116 (1994) 257–264. https://doi.org/10.1115/1.2899218.
[51] G.R. Jonsson, S. Lalot, O.P. Palsson, B. Desmet, Use of extended Kalman filtering in detecting fouling in heat exchangers, International Journal of Heat and Mass Transfer 50 (2007) 2643–2655. https://doi.org/10.1016/j.ijheatmasstransfer.2006.11.025.
[52] R.M. Paruthipulli Kalarikkal, J. Subbaiah, Modified Dual EKF with Machine Learning Model for Fouling Prediction of Industrial Heat Exchanger, Korean J. Chem. Eng. (2024). https://doi.org/10.1007/s11814-024-00128-y.
[53] A.K. Sivathanu, S. Subramanian, Extended Kalman filter for fouling detection in thermal power plant reheater, Control Engineering Practice 73 (2018) 91–99. https://doi.org/10.1016/j.conengprac.2018.01.005.
[54] A.K. Sivathanu, K. Vaidyanathan, N. Murugan, Extended Kalman filter based tube leak detection for thermal power plant reheater, AIP Conference Proceedings 2813 (2023) 020001. https://doi.org/10.1063/5.0156886.
[55] S.P.K. Ayyagari, R. Singh, P. Purkayastha, P. Deeskow, Estimation of soot deposition in the boilers of coal fired power plant, SYMOPA, Trivandrum (2010) 362–367.
[56] H. Noormohamadi, M. Suresrafil, The unscented Kalman Filter for the Estimation the States of The Boiler-Turbin Model, International Journal of Scientific & Engineering Research 4 (2013). https://www.ijser.org/researchpaper/the-unscented-kalman-filter-for-the-estimation-the-states-of-the-boiler-turbin-model.pdf.
[57] M. Hultgren, E. Ikonen, J. Kovacs, Circulating fluidized bed boiler state estimation with an unscented Kalman filter tool, in: 2014 IEEE Conference on Control Applications (CCA), IEEE, Juan Les Antibes, France, 2014: pp. 310–315. https://doi.org/10.1109/CCA.2014.6981364.
[58] C.S. Somayajula, D. Bhattacharyya, S. Hu, X. Liu, Hot Corrosion Monitoring in Boilers Using a Nonlinear Estimator and Electrochemical Noise-Based Corrosion Sensors, Ind. Eng. Chem. Res. 63 (2024) 394–409. https://doi.org/10.1021/acs.iecr.3c03542.
[59] R. Kumar Mandela, R. Rengaswamy, S. Narasimhan, L.N. Sridhar, Recursive state estimation techniques for nonlinear differential algebraic systems, Chemical Engineering Science 65 (2010) 4548–4556. https://doi.org/10.1016/j.ces.2010.04.020.
[60] S. Bhase, M. Bhushan, S. Kadu, S. Mukhopadhyay, Continuous–discrete filtering techniques for estimating states of nonlinear differential–algebraic equations (DAEs) systems, Int. J. Dynam. Control 11 (2023) 162–182. https://doi.org/10.1007/s40435-022-00955-z.

[61] V.M. Becerra, P.D. Roberts, G.W. Griffiths, Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations, Control Engineering Practice 9 (2001) 267–281. https://doi.org/10.1016/S0967-0661(00)00110-6.
[62] P. Mobed, S. Munusamy, D. Bhattacharyya, R. Rengaswamy, State and Parameter Estimation in Distributed Constrained Systems. 1. Extended Kalman Filtering of a Special Class of Differential-Algebraic Equation Systems, Ind. Eng. Chem. Res. 56 (2017) 206–215. https://doi.org/10.1021/acs.iecr.6b02796.
[63] V. Saini, Dynamic Modeling and Estimation for Condition Monitoring of Energy Systems, in: 2024 AIChE Annual Meeting, AIChE, 2024. https://aiche.confex.com/aiche/2024/meetingapp.cgi/Paper/697039.
[64] F. Alobaid, N. Mertens, R. Starkloff, T. Lanz, C. Heinze, B. Epple, Progress in dynamic simulation of thermal power plants, Progress in Energy and Combustion Science 59 (2017) 79–162. https://doi.org/10.1016/j.pecs.2016.11.001.
[65] R.B. Bird, W.E. Stewart, E.N. Lightfoot, Transport phenomena, Revised ed, Wiley, New York, 2007.
[66] K. Hedrick, E. Hedrick, B. Omell, S.E. Zitney, D. Bhattacharyya, Dynamic Modeling, Parameter Estimation, and Data Reconciliation of a Supercritical Pulverized Coal-Fired Boiler, Ind. Eng. Chem. Res. 61 (2022) 16764–16779. https://doi.org/10.1021/acs.iecr.2c01977.
[67] H.-J. Kretzschmar, W. Wagner, IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam, in: H.-J. Kretzschmar, W. Wagner (Eds.), International Steam Tables: Properties of Water and Steam Based on the Industrial Formulation IAPWS-IF97, Springer, Berlin, Heidelberg, 2019: pp. 7–150. https://doi.org/10.1007/978-3-662-53219-5_3.
[68] S.S. Haykin, ed., Kalman filtering and neural networks, Wiley, New York, 2001.
[69] Y. Wang, D. Bhattacharyya, R. Turton, Multiobjective Dynamic Optimization for Optimal Load-Following of Natural Gas Combined Cycle Power Plants under Stress Constraints, Ind. Eng. Chem. Res. 60 (2021) 14251–14270. https://doi.org/10.1021/acs.iecr.1c01461.
[70] K. Hedrick, B. Omell, S.E. Zitney, D. Bhattacharyya, Development of a health monitoring framework: Application to a supercritical pulverized coal-fired boiler, Energy 290 (2024) 130153. https://doi.org/10.1016/j.energy.2023.130153.
[71] V. Venkatasubramanian, The promise of artificial intelligence in chemical engineering: Is it here, finally?, AIChE Journal 65 (2019) 466–478. https://doi.org/10.1002/aic.16489.
[72] A. Mukherjee, D. Bhattacharyya, Hybrid Series/Parallel All-Nonlinear Dynamic-Static Neural Networks: Development, Training, and Application to Chemical Processes, Industrial and Engineering Chemistry Research (2023). https://doi.org/10.1021/acs.iecr.2c03339.
[73] J.C. Hoskins, K.M. Kaliyur, D.M. Himmelblau, Fault Diagnosis in Complex Chemical Plants Using Artificial Neural Networks, 37 (1991) 137–141.
[74] T. Poggio, H. Mhaskar, B. Miranda, Why and When Can Deep-but Not Shallow-networks Avoid the Curse of Dimensionality : A Review, (n.d.) 1–17. https://doi.org/10.1007/s11633-017-1054-2.
[75] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Reviews Physics 3 (2021) 422–440. https://doi.org/10.1038/s42254-021-00314-5.
[76] M.M. Bejani, M. Ghatee, A systematic review on overfitting control in shallow and deep neural networks, Springer Netherlands, 2021. https://doi.org/10.1007/s10462-021-09975-1.
[77] Y.M. Chiang, L.C. Chang, F.J. Chang, Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modeling, Journal of Hydrology 290 (2004) 297–311. https://doi.org/10.1016/j.jhydrol.2003.12.033.

[78] D. Bertsimas, R. Shioda, Algorithm for cardinality-constrained quadratic optimization, Computational Optimization and Applications 43 (2009) 1–22. https://doi.org/10.1007/s10589-007-9126-9.
[79] H. Liu, H. Motoda, Computational Methods of Feature Selection, Chapman & Hal/CRC, London, England, UK, 2007. https://doi.org/10.1201/9781584888796.
[80] K. McBride, K. Sundmacher, Overview of Surrogate Modeling in Chemical Process Engineering, Chemie-Ingenieur-Technik 91 (2019) 228–239. https://doi.org/10.1002/cite.201800091.
[81] T.W. Simpson, T.M. Mauery, J. Korte, F. Mistree, Kriging models for global approximation in simulation-based multidisciplinary design optimization, AIAA Journal 39 (2001) 2233–2241. https://doi.org/10.2514/3.15017.
[82] A.I.J. Forrester, A. Sóbester, A.J. Keane, Engineering Design via Surrogate Modelling: A Practical Guide, John Wiley & Sons Ltd, UK., 2008. https://doi.org/10.1002/9780470770801.
[83] F. Boukouvala, F.J. Muzzio, M.G. Ierapetritou, Dynamic data-driven modeling of pharmaceutical processes, Industrial and Engineering Chemistry Research 50 (2011) 6743–6754. https://doi.org/10.1021/ie102305a.
[84] J.H.A. Guillaume, J.D. Jakeman, S. Marsili-Libelli, M. Asher, P. Brunner, B. Croke, M.C. Hill, A.J. Jakeman, K.J. Keesman, S. Razavi, J.D. Stigter, Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose, Environmental Modelling and Software 119 (2019) 418–432. https://doi.org/10.1016/j.envsoft.2019.07.007.
[85] Z.T. Wilson, N. V. Sahinidis, The ALAMO approach to machine learning, Computers and Chemical Engineering 106 (2017) 785–795. https://doi.org/10.1016/j.compchemeng.2017.02.010.
[86] A. Bhosekar, M. Ierapetritou, Advances in surrogate based modeling, feasibility analysis, and optimization: A review, Computers and Chemical Engineering 108 (2018) 250–267. https://doi.org/10.1016/j.compchemeng.2017.09.017.
[87] A. Cozad, N. V. Sahinidis, D.C. Miller, A combined first-principles and data-driven approach to model building, Computers and Chemical Engineering 73 (2015) 116–127. https://doi.org/10.1016/j.compchemeng.2014.11.010.
[88] A. Cozad, N. V. Sahinidis, D.C. Miller, Learning Surrogate Models for Simulation-Based Optimization, AIChE J 60 (2014).
[89] Z.T. Wilson, N. V. Sahinidis, The ALAMO approach to machine learning, Computers and Chemical Engineering 106 (2017) 785–795. https://doi.org/10.1016/j.compchemeng.2017.02.010.
[90] L. Ljung, T. Glad, Modeling of Dynamic Systems, 1994th ed., Prentice Hall; Englewood Cliffs, NJ, USA, 1994.
[91] B. Bhadriraju, M.S.F. Bangi, A. Narasingam, J.S. Il Kwon, Operable adaptive sparse identification of systems: Application to chemical processes, AIChE Journal 66 (2020) 1–15. https://doi.org/10.1002/aic.16980.
[92] N.M. Mangan, J.N. Kutz, S.L. Brunton, J.L. Proctor, Model selection for dynamical systems via sparse regression and information criteria, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473 (2017). https://doi.org/10.1098/rspa.2017.0009.
[93] W. Sun, R.D. Braatz, ALVEN : Algebraic learning via elastic net for static and dynamic nonlinear model identification, Computers and Chemical Engineering 143 (2020) 107103. https://doi.org/10.1016/j.compchemeng.2020.107103.
[94] P. Gelß, S. Klus, J. Eisert, C. Schütte, Multidimensional Approximation of Nonlinear Dynamical Systems, Journal of Computational and Nonlinear Dynamics 14 (2019). https://doi.org/10.1115/1.4043148.

[95] M. Schmidt, H. Lipson, Distilling Free-Form Natural Laws, 324 (2009) 81–86.
[96] W. Sun, R.D. Braatz, ALVEN : Algebraic learning via elastic net for static and dynamic nonlinear model identification, Computers and Chemical Engineering 143 (2020) 107103. https://doi.org/10.1016/j.compchemeng.2020.107103.
[97] S.L. Brunton, J.L. Proctor, J.N. Kutz, W. Bialek, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proceedings of the National Academy of Sciences of the United States of America 113 (2016) 3932–3937. https://doi.org/10.1073/pnas.1517384113.
[98] J.C. Loiseau, S.L. Brunton, Constrained sparse Galerkin regression, Journal of Fluid Mechanics 838 (2018) 42–67. https://doi.org/10.1017/jfm.2017.823.
[99] R.R. Nadakuditi, J.W. Silverstein, Fundamental limit of sample generalized eigenvalue based detection of signals in noise using relatively few signal-bearing and noise-only samples, IEEE Journal on Selected Topics in Signal Processing 4 (2010) 468–480. https://doi.org/10.1109/JSTSP.2009.2038310.
[100] B. Bhadriraju, J.S. Il Kwon, F. Khan, Risk-based fault prediction of chemical processes using operable adaptive sparse identification of systems (OASIS), Computers and Chemical Engineering 152 (2021) 107378. https://doi.org/10.1016/j.compchemeng.2021.107378.
[101] B. Bhadriraju, J.S. Kwon, F. Khan, OASIS-P: Operable Adaptive Sparse Identification of Systems for fault Prognosis of chemical processes, Journal of Process Control 107 (2021) 114–126.
[102] E. Kaiser, J.N. Kutz, S.L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474 (2018). https://doi.org/10.1098/rspa.2018.0335.
[103] F. Abdullah, Z. Wu, P.D. Christofides, Sparse-identification-based model predictive control of nonlinear two-time-scale processes, Computers and Chemical Engineering 153 (2021) 107411. https://doi.org/10.1016/j.compchemeng.2021.107411.
[104] K. Kaheman, J.N. Kutz, S.L. Brunton, SINDy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics: SINDy-PI, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 476 (2020). https://doi.org/10.1098/rspa.2020.0279.
[105] Z. Ghahramani, S.T. Roweis, Learning nonlinear dynamical systems using an EM algorithm, Advances in Neural Information Processing Systems (1999) 431–437.
[106] B. Chabane, B. Daoued, On the use of Kalman filter for enhancing speech corrupted by colored noise, WSEAS Transactions on Signal Processing 4 (2008) 657–666.
[107] X. Jin, B. Huang, Robust identification of piecewise/switching autoregressive exogenous process, AIChE Journal 56 (2010). https://doi.org/10.1002/aic.12112.
[108] R.B. Gopaluni, A particle filter approach to identification of nonlinear processes under missing observations, Canadian Journal of Chemical Engineering 86 (2008) 1081–1092. https://doi.org/10.1002/cjce.20113.
[109] Y. Huang, Y. Zhang, N. Li, S. Mohsen Naqvi, J. Chambers, A robust and efficient system identification method for a state-space model with heavy-tailed process and measurement noises, FUSION 2016 - 19th International Conference on Information Fusion, Proceedings (2016) 441–448.
[110] F. Guo, K. Hariprasad, B. Huang, Y.S. Ding, Robust identification for nonlinear errors-in-variables systems using the EM algorithm, Journal of Process Control 54 (2017) 129–137. https://doi.org/10.1016/j.jprocont.2017.03.008.

[111] O. Osoba, B. Kosko, Noisy Expectation-Maximization: Applications and Generalizations, (2018) 1–5.
[112] R. Pradeepa, G. V. Anand, Estimation of signals in colored non Gaussian noise based on Gaussian mixture models, NSSPW - Nonlinear Statistical Signal Processing Workshop 2006 (2006) 17–20. https://doi.org/10.1109/NSSPW.2006.4378810.
[113] T.B. Schön, A. Wills, B. Ninness, System identification of nonlinear state-space models, Automatica 47 (2011) 39–49. https://doi.org/10.1016/j.automatica.2010.10.013.
[114] Y. Huang, Y. Zhang, N. Li, S. Mohsen Naqvi, J. Chambers, A robust and efficient system identification method for a state-space model with heavy-tailed process and measurement noises, FUSION 2016 - 19th International Conference on Information Fusion, Proceedings (2016) 441–448.
[115] O. Osoba, B. Kosko, Noisy Expectation-Maximization: Applications and Generalizations, (2018) 1–5.
[116] R. Pradeepa, G. V. Anand, Estimation of signals in colored non Gaussian noise based on Gaussian mixture models, NSSPW - Nonlinear Statistical Signal Processing Workshop 2006 (2006) 17–20. https://doi.org/10.1109/NSSPW.2006.4378810.
[117] T. Bankole, D. Bhattacharyya, Exploiting connectivity structures for decomposing process plants, Journal of Process Control 71 (2018) 116–129. https://doi.org/10.1016/j.jprocont.2018.09.002.
[118] P. Buchwald, A general bilinear model to describe growth or decline time profiles, 205 (2007) 108–136. https://doi.org/10.1016/j.mbs.2006.08.013.
[119] T. Doghri, L. Szczecinski, J. Benesty, A. Mitiche, Bilinear Models for Machine Learning, (n.d.).
[120] K.J. Friston, L. Harrison, W. Penny, Dynamic causal modelling, NeuroImage 19 (2003) 1273–1302. https://doi.org/10.1016/S1053-8119(03)00202-7.
[121] K. Lee, E.N. Johnson, State estimation using Gaussian process regression for colored noise systems, IEEE Aerospace Conference Proceedings (2017). https://doi.org/10.1109/AERO.2017.7943781.
[122] A.E. Bryson, L.J. Henrikson, Estimation using sampled data containing sequentially correlated noise, Journal of Spacecraft and Rockets 5 (1968) 662–665. https://doi.org/10.2514/3.29327.
[123] G. Chang, On kalman filter for linear system with colored measurement noise, Journal of Geodesy 88 (2014) 1163–1170. https://doi.org/10.1007/s00190-014-0751-7.
[124] A.H. Land, A.G. Doig, An Automatic Method of Solving Discrete Programming Problems, Econometrica 28 (1960) 497–520.
[125] D. Bienstock, Computational study of a family of mixed-integer quadratic programming problems, Mathematical Programming 74 (1996) 121–140.
[126] D. Jones, D. Bhattacharyya, R. Turton, S.E. Zitney, Plant-wide control system design: Primary controlled variable selection, Computers and Chemical Engineering 71 (2014) 220–234. https://doi.org/10.1016/j.compchemeng.2014.08.004.
[127] T. Bankole, D. Bhattacharyya, P. Pezzini, B. Gebreslassie, N.F. Harun, D. Tucker, U. Diwekar, K.M. Bryden, Multiobjective Optimal Controlled Variable Selection for a Gas Turbine-Solid Oxide Fuel Cell System Using a Multiagent Optimization Platform, Industrial and Engineering Chemistry Research 59 (2020) 20058–20070. https://doi.org/10.1021/acs.iecr.0c02865.
[128] T. Bankole, P. Pezzini, D. Tucker, K. Bryden, D. Bhattacharyya, Optimal controlled variable selection for cyber-physical systems, American Society of Mechanical Engineers, Power Division (Publication) POWER 1 (2018). https://doi.org/10.1115/POWER2018-7486.
[129] J.M. Ferguson, M.L. Taper, R. Zenil-Ferguson, M. Jasieniuk, B.D. Maxwell, Incorporating Parameter Estimability Into Model Selection, Frontiers in Ecology and Evolution 7 (2019) 1–15. https://doi.org/10.3389/fevo.2019.00427.

[130] H.A.I. Aike, A New Look at the Statistical Model Identification, IEEE Transactions on Automatic Control 19, No 6 (1974) 716–723.
[131] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum Likelihood from Incomplete Data Via the EM Algorithm , Journal of the Royal Statistical Society: Series B (Methodological) 39 (1977) 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.
[132] A.P. Dempster, D.B. Rubin, R.K. Tsutakawa, Estimation in covariance components models, Journal of the American Statistical Association 76 (1981) 341–353. https://doi.org/10.1080/01621459.1981.10477653.
[133] K.J. Friston, W. Penny, C. Phillips, S. Kiebel, G. Hinton, J. Ashburner, Classical and Bayesian inference in neuroimaging: Theory, NeuroImage 16 (2002) 465–483. https://doi.org/10.1006/nimg.2002.1090.
[134] K.J. Friston, L. Harrison, W. Penny, Dynamic causal modelling, NeuroImage 19 (2003) 1273–1302. https://doi.org/10.1016/S1053-8119(03)00202-7.
[135] J.G. VandeVusse, Plug-flow type reactor versus tank reactor, Chemical Engineering Science 19 (1964) 994–997.
[136] K.U. Klatt, S. Engell, Gain-scheduling trajectory control of a continuous stirred tank reactor, Computers and Chemical Engineering 22 (1998) 491–502. https://doi.org/10.1016/S0098-1354(97)00261-5.
[137] S. Adeyemo, D. Bhattacharyya, Optimal nonlinear dynamic sparse model selection and Bayesian parameter estimation for nonlinear systems, Computers and Chemical Engineering 180 (2024) 108502. https://doi.org/10.1016/j.compchemeng.2023.108502.
[138] A.S. Chinen, J.C. Morgan, B. Omell, D. Bhattacharyya, D.C. Miller, Dynamic Data Reconciliation and Validation of a Dynamic Model for Solvent-Based CO 2 Capture Using Pilot-Plant Data, Industrial and Engineering Chemistry Research 58 (2019) 1978–1993. https://doi.org/10.1021/acs.iecr.8b04489.
[139] J.C. Morgan, D. Bhattacharyya, C. Tong, M. David, Uncertainty Quantification of Property Models: Methodology and Its Application to CO2-Loaded Aqueous MEA Solutions, AIChE Journal 61 (2015) 1822–1839.
[140] J.C. Morgan, A.S. Chinen, B. Omell, D. Bhattacharyya, C. Tong, D.C. Miller, Thermodynamic modeling and uncertainty quantification of CO2-loaded aqueous MEA solutions, Chemical Engineering Science 168 (2017) 309–324. https://doi.org/10.1016/j.ces.2017.04.049.
[141] J.C. Morgan, A. Soares Chinen, B. Omell, D. Bhattacharyya, C. Tong, D.C. Miller, B. Buschle, M. Lucquiaud, Development of a Rigorous Modeling Framework for Solvent-Based CO2 Capture. Part 2: Steady-State Validation and Uncertainty Quantification with Pilot Plant Data, Industrial and Engineering Chemistry Research 57 (2018) 10464–10481. https://doi.org/10.1021/acs.iecr.8b01472.
[142] B. Bhadriraju, M.S.F. Bangi, A. Narasingam, J.S. Il Kwon, Operable adaptive sparse identification of systems: Application to chemical processes, AIChE Journal 66 (2020) 1–15. https://doi.org/10.1002/aic.16980.
[143] U. Fasel, J.N. Kutz, B.W. Brunton, S.L. Brunton, Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 478 (2022). https://doi.org/10.1098/rspa.2021.0904.
[144] F. Abdullah, P.D. Christofides, Data-based modeling and control of nonlinear process systems using sparse identification: An overview of recent results, Computers and Chemical Engineering 174 (2023) 108247. https://doi.org/10.1016/j.compchemeng.2023.108247.

[145] F. Abdullah, M.S. Alhajeri, P.D. Christofides, Modeling and Control of Nonlinear Processes Using Sparse Identification : Using Dropout to Handle Noisy Data, (2022). https://doi.org/10.1021/acs.iecr.2c02639.
[146] F. Abdullah, Z. Wu, P.D. Christofides, Handling noisy data in sparse model identification using subsampling and co-teaching, Computers and Chemical Engineering 157 (2022) 107628. https://doi.org/10.1016/j.compchemeng.2021.107628.
[147] V. Saini, D. Bhattacharyya, D. Purdy, J. Parker, C. Boohaker, Nonlinear state estimation of a power plant superheater by using the extended Kalman filter for differential algebraic equation systems, Applied Thermal Engineering 251 (2024) 123471. https://doi.org/10.1016/j.applthermaleng.2024.123471.
[148] J. Sansana, M.N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L.H. Chiang, M.S. Reis, Recent trends on hybrid modeling for Industry 4.0, Computers and Chemical Engineering 151 (2021) 107365. https://doi.org/10.1016/j.compchemeng.2021.107365.
[149] W. Bradley, J. Kim, Z. Kilwein, L. Blakely, M. Eydenberg, J. Jalvin, C. Laird, F. Boukouvala, Perspectives on the integration between first-principles and data-driven modeling, Computers and Chemical Engineering 166 (2022) 107898. https://doi.org/10.1016/j.compchemeng.2022.107898.
[150] T. Asrav, E. Aydin, Physics-informed recurrent neural networks and hyper-parameter optimization for dynamic process systems, Computers and Chemical Engineering 173 (2023) 108195. https://doi.org/10.1016/j.compchemeng.2023.108195.
[151] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.
[152] Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, Physics-Informed Machine Learning : A Survey on Problems , Methods and Applications, (2022) 1–45.
[153] S. Cuomo, V. Schiano, D. Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific Machine Learning Through Physics – Informed Neural Networks : Where we are and What ' s Next, Journal of Scientific Computing 92 (2022) 1–62. https://doi.org/10.1007/s10915-022-01939-z.
[154] L. Yang, X. Meng, G.E. Karniadakis, B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, Journal of Computational Physics 425 (2021) 1–32. https://doi.org/10.1016/j.jcp.2020.109913.
[155] W.L. Buntine, A.S. Weigen, Bayesian Back-Propagation, Complex Systems 5 (1991) 603–643.
[156] R.M. Neal, Bayesian training of backpropagation networks by the hybrid Monte Carlo method, Dept. of Computer Science, University of Toronto, Tech. … (1992) 1–21.
[157] D.J.C. MacKay, A Practical Bayesian Framework, Neural Computation 472 (1992) 448–472.
[158] F.G. De Freitas, Bayesian Methods for Neural Networks, University of Cambridge, 1999.
[159] J.F.G. De Freitas, M. Niranjan, A.H. Gee, Dynamic learning with the EM algorithm for neural networks, Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology 26 (2000) 119–131. https://doi.org/10.1023/a:1008103718973.
[160] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, Journal of Computational Physics 394 (2019) 56–81. https://doi.org/10.1016/j.jcp.2019.05.024.
[161] A. Mukherjee, D. Bhattacharyya, On the Development of Steady-State and Dynamic Mass-Constrained Neural Networks Using Noisy Transient Data, Computers & Chemical Engineering 187 (2024) 108722. https://doi.org/10.1016/j.compchemeng.2024.108722.

[162] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, 2006. https://doi.org/10.1007/s10107-004-0559-y.
[163] J.C. Morgan, A.S. Chinen, B. Omell, D. Bhattacharyya, C. Tong, D.C. Miller, Thermodynamic modeling and uncertainty quantification of CO2-loaded aqueous MEA solutions, Chemical Engineering Science 168 (2017) 309–324. https://doi.org/10.1016/j.ces.2017.04.049.
[164] V. Venkatasubramanian, The promise of artificial intelligence in chemical engineering: Is it here, finally?, AIChE Journal 65 (2019) 466–478. https://doi.org/10.1002/aic.16489.
[165] J. Panerati, M.A. Schnellmann, C. Patience, G. Beltrame, G.S. Patience, Experimental methods in chemical engineering: Artificial neural networks–ANNs, The Canadian Journal of Chemical Engineering 97 (2019) 2372–2382. https://doi.org/10.1002/cjce.23507.
[166] A. Mukherjee, D. Bhattacharyya, Hybrid Series/Parallel All-Nonlinear Dynamic-Static Neural Networks: Development, Training, and Application to Chemical Processes, Industrial and Engineering Chemistry Research 62 (2023) 3221–3237. https://doi.org/10.1021/acs.iecr.2c03339.
[167] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.
[168] A. Mukherjee, D. Bhattacharyya, On the Development of Steady-State and Dynamic Mass-Constrained Neural Networks Using Noisy Transient Data, Computers & Chemical Engineering 187 (2024) 108722. https://doi.org/10.1016/j.compchemeng.2024.108722.
[169] A. Mukherjee, D. Bhattacharyya, Development of Steady-State and Dynamic Mass and Energy Constrained Neural Networks for Distributed Chemical Systems Using Noisy Transient Data, Industrial & Engineering Chemistry Research (2024). https://doi.org/10.1021/acs.iecr.4c01429.
[170] A. Mukherjee, D. Bhattacharyya, Development of Steady-State and Dynamic Mass-Energy Constrained Neural Networks using Noisy Transient Data, in: CSChE Systems & Control Transactions, PSE Press, 2024: pp. 330–337.
[171] H. Wang, Y. Oh, E.S. Yoon, Strategies for modeling and control of nonlinear chemical processes using neural networks, Computers and Chemical Engineering 22 (1998) 0–3. https://doi.org/10.1016/s0098-1354(98)00157-4.
[172] T. Xie, S.M. Ghiaasiaan, S. Karrila, Artificial neural network approach for flow regime classification in gas-liquid-fiber flows based on frequency domain analysis of pressure signals, Chemical Engineering Science 59 (2004) 2241–2251. https://doi.org/10.1016/j.ces.2004.02.017.
[173] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, K. Yin, A review of process fault detection and diagnosis, Computers & Chemical Engineering 27 (2003) 327–346. https://doi.org/10.1016/s0098-1354(02)00162-x.
[174] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S.N. Kavuri, A review of fault detection and diagnosis. Part III: Process history based methods, Computers and Chemical Engineering 27 (2003) 327–346.
[175] S.A. Kalogirou, Artificial intelligence for the modeling and control of combustion processes: A review, 2003. https://doi.org/10.1016/S0360-1285(03)00058-3.
[176] D.M. Himmelblau, Accounts of experiences in the application of artificial neural networks in chemical engineering, Industrial and Engineering Chemistry Research 47 (2008) 5782–5796. https://doi.org/10.1021/ie800076s.
[177] D.W. Coit, B.T. Jackson, A.E. Smith, Static neural network process models: Considerations and case studies, International Journal of Production Research 36 (1998) 2953–2967. https://doi.org/10.1080/002075498192229.

[178] J. Zupan, J. Gasteiger, Neural networks: A new method for solving chemical problems or just a passing phase?, Analytica Chimica Acta 248 (1991) 1–30. https://doi.org/10.1016/S0003-2670(00)80865-X.
[179] J. Manikandan, K.Y. Naidu, B. Janardhan, M.R. Kishore, K. Prabhakar, Comparison of static and dynamic neural network models in predicting outlet temperature of shell and tube heat exchanger, International Journal of Vehicle Structures and Systems 11 (2019) 585–588. https://doi.org/10.4273/ijvss.11.5.21.
[180] A. El-shafie, Dynamic versus static neural network model for rainfall forecasting at Klang River Basin, Malaysia, Hydrology and Earth System Sciences Discussions 8 (2011) 6489–6532. https://doi.org/10.5194/hessd-8-6489-2011.
[181] A. Poznyak, I. Chairez, T. Poznyak, A survey on artificial neural networks application for identification and control in environmental engineering: Biological and chemical systems with uncertain models, Annual Reviews in Control 48 (2019) 250–272. https://doi.org/10.1016/j.arcontrol.2019.07.003.
[182] E. Pisoni, M. Farina, C. Carnevale, L. Piroddi, Forecasting peak air pollution levels using NARX models, Engineering Applications of Artificial Intelligence 22 (2009) 593–602. https://doi.org/10.1016/j.engappai.2009.04.002.
[183] L. Ruiz, M. Cuéllar, M. Calvo-Flores, M. Jiménez, An Application of Non-Linear Autoregressive Neural Networks to Predict Energy Consumption in Public Buildings, Energies 9 (2016) 684. https://doi.org/10.3390/en9090684.
[184] P. Guo, Nonlinear predictive functional control based on hopfield network and its application in CSTR, Proceedings of the 2006 International Conference on Machine Learning and Cybernetics 2006 (2006) 3036–3039. https://doi.org/10.1109/ICMLC.2006.258361.
[185] M.N. Kashani, S. Shahhosseini, A methodology for modeling batch reactors using generalized dynamic neural networks, Chemical Engineering Journal 159 (2010) 195–202. https://doi.org/10.1016/j.cej.2010.02.053.
[186] T. Ganesan, P. Vasant, I. Elamvazuthi, Hopfield neural networks approach for design optimization of hybrid power systems with multiple renewable energy sources in a fuzzy environment, Journal of Intelligent & Fuzzy Systems 26 (2014) 2143–2154. https://doi.org/10.3233/IFS-130889.
[187] W.K. Tsinghua, D. Huang, F. Yang, Y. Jiang, Soft sensor development and applications based on LSTM in deep neural networks, 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings 2018-Janua (2018) 1–6. https://doi.org/10.1109/SSCI.2017.8280954.
[188] P. Roussel-Ragot, L. Personnaz, G. Dreyfus, S. Marcos, Communicated by Steven Nowlan Neural Networks and Nonlinear Adaptive Filtering: Unifying Concepts and New Algorithms 0. Nerrand, 199 (1993) 165–199.
[189] K. Schädler, F. Wysotzki, Comparing structures using a Hopfield-style neural network, Applied Intelligence 11 (1999) 15–30. https://doi.org/10.1023/A:1008320413168.
[190] M. Ayoubi, Comparison between the dynamic multi-layered perceptron and the generalised Hammerstein model for experimental identification of the loading process in diesel engines, Control Engineering Practice 6 (1998) 271–279. https://doi.org/10.1016/S0967-0661(98)00004-5.
[191] Y.M. Chiang, L.C. Chang, F.J. Chang, Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modeling, Journal of Hydrology 290 (2004) 297–311. https://doi.org/10.1016/j.jhydrol.2003.12.033.
[192] M.M. Bejani, M. Ghatee, A systematic review on overfitting control in shallow and deep neural networks, Springer Netherlands, 2021. https://doi.org/10.1007/s10462-021-09975-1.

[193] M. Hong, S. Cheng, Model predictive control based on Kalman filter for constrained hammerstein-wiener systems, Mathematical Problems in Engineering 2013 (2013). https://doi.org/10.1155/2013/104702.
[194] S. Mahmoodi, J. Poshtan, M.R. Jahed-Motlagh, A. Montazeri, Nonlinear model predictive control of a pH neutralization process based on Wiener-Laguerre model, Chemical Engineering Journal 146 (2009) 328–337. https://doi.org/10.1016/j.cej.2008.06.010.
[195] G. Shafiee, M.M. Arefi, M.R. Jahed-Motlagh, A.A. Jalali, Nonlinear predictive control of a polymerization reactor based on piecewise linear Wiener model, Chemical Engineering Journal 143 (2008) 282–292. https://doi.org/10.1016/j.cej.2008.05.013.
[196] S.I. Biagiola, J.L. Figueroa, Identification of uncertain MIMO Wiener and Hammerstein models, Computers and Chemical Engineering 35 (2011) 2867–2875. https://doi.org/10.1016/j.compchemeng.2011.05.013.
[197] R. Kanthasamy, H. Anwaruddin, S.K. Sinnadurai, A new approach to the identification of distillation column based on Hammerstein model, Modelling and Simulation in Engineering 2014 (2014). https://doi.org/10.1155/2014/813757.
[198] H. Zhao, J. Guiver, R. Neelakantan, L.T. Biegler, A nonlinear industrial model predictive controller using integrated PLS and neural net state-space model, Control Engineering Practice 9 (2001) 125–133. https://doi.org/10.1016/S0967-0661(00)00101-5.
[199] C.D. Kappatou, D. Bongartz, J. Najman, S. Sass, A. Mitsos, Global dynamic optimization with Hammerstein–Wiener models embedded, Journal of Global Optimization (2022). https://doi.org/10.1007/s10898-022-01145-z.
[200] T. Wigren, Recursive prediction error identification using the nonlinear wiener model, Automatica 29 (1993) 1011–1025. https://doi.org/10.1016/0005-1098(93)90103-Z.
[201] M. Schoukens, A. Marconato, R. Pintelon, G. Vandersteen, Y. Rolain, Parametric identification of parallel Wiener-Hammerstein systems, Automatica 51 (2015) 111–122. https://doi.org/10.1016/j.automatica.2014.10.105.
[202] P. Saha, S.H. Krishnan, V.S.R. Rao, S.C. Patwardhan, Modeling and predictive control of MIMO nonlinear systems using Wiener-Laguerre models, Chemical Engineering Communications 191 (2004) 1083–1119. https://doi.org/10.1080/00986440492764 52.
[203] S.L. Lacy, D.S. Bernstein, Identification of FIR Wiener systems with unknown, non-invertible, polynomial non-linearities, International Journal of Control 76 (2003) 1500–1507. https://doi.org/10.1080/00207170310001599122.
[204] J. Ma, P. Mahapatra, S.E. Zitney, L.T. Biegler, D.C. Miller, D-RM Builder: A software tool for generating fast and accurate nonlinear dynamic reduced models from high-fidelity models, Computers and Chemical Engineering 94 (2016) 60–74. https://doi.org/10.1016/j.compchemeng.2016.07.021.
[205] G.B. Sentoni, L.T. Biegler, J.P. Guiver, Model reduction for nonlinear DABNet models, Proceedings of the American Control Conference 3 (1999) 2052–2056. https://doi.org/10.1109/acc.1999.786278.
[206] B.P. Omell, J. Ma, P. Mahapatra, M. Yu, A. Lee, D. Bhattacharyya, S.E. Zitney, L.T. Biegler, D.C. Miller, Advanced Modeling and Control of a Solid Sorbent-Based CO2 Capture Process, IFAC-PapersOnLine 49 (2016) 633–638. https://doi.org/10.1016/j.ifacol.2016.07.240.
[207] Y. Yu, X. Si, C. Hu, J. Zhang, A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures, Neural Computation 31 (2019) 1235–1270. https://doi.org/10.1162/neco_a_01199.

[208] Y.M. Ren, M.S. Alhajeri, J. Luo, S. Chen, F. Abdullah, Z. Wu, P.D. Christofides, A tutorial review of neural network modeling approaches for model predictive control, Computers and Chemical Engineering 165 (2022) 107956. https://doi.org/10.1016/j.compchemeng.2022.107956.
[209] E. Esche, J. Weigert, G. Brand Rihm, J. Göbel, J.-U. Repke, Architectures for neural networks as surrogates for dynamic systems in chemical engineering, Chemical Engineering Research and Design 177 (2022) 184–199. https://doi.org/10.1016/j.cherd.2021.10.042.
[210] M. Schoukens, K. Tiels, Identification of block-oriented nonlinear systems starting from linear approximations: A survey, Automatica 85 (2017) 272–292. https://doi.org/10.1016/j.automatica.2017.06.044.
[211] L. Chen, Y. Hontoir, D. Huang, J. Zhang, A.J. Morris, Combining first principles with black-box techniques for reaction systems, Control Engineering Practice 12 (2004) 819–826. https://doi.org/10.1016/j.conengprac.2003.09.006.
[212] K. Hammar, T. Djamah, M. Bettayeb, Nonlinear system identification using fractional Hammerstein–Wiener models, Nonlinear Dynamics 98 (2019) 2327–2338. https://doi.org/10.1007/s11071-019-05331-9.
[213] A. Lazrak, F. Boudehenn, S. Bonnot, G. Fraisse, A. Leconte, P. Papillon, B. Souyri, Development of a dynamic artificial neural network model of an absorption chiller and its experimental validation, Renewable Energy 86 (2016) 1009–1022. https://doi.org/10.1016/j.renene.2015.09.023.
[214] J.M. Ferguson, M.L. Taper, R. Zenil-Ferguson, M. Jasieniuk, B.D. Maxwell, Incorporating Parameter Estimability Into Model Selection, Frontiers in Ecology and Evolution 7 (2019) 1–15. https://doi.org/10.3389/fevo.2019.00427.
[215] K.-L. Du, M.N.S. Swamy, Neural Networks and Statistical Learning, Springer London, London, 2014. https://doi.org/10.1007/978-1-4471-5571-3.
[216] Y. Huang, Advances in artificial neural networks - Methodological development and application, Algorithms 2 (2009) 973–1007. https://doi.org/10.3390/algor2030973.
[217] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, Y. Wang, Dynamic Neural Networks: A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 8828 (2021) 1–20. https://doi.org/10.1109/TPAMI.2021.3117837.
[218] B.A. Pearlmutter, Gradient Calculations for Dynamic Recurrent Neural Networks: A Survey, IEEE Transactions on Neural Networks 6 (1995) 1212–1228. https://doi.org/10.1109/72.410363.
[219] O. De Jesús, M.T. Hagan, Backpropagation algorithms for a broad class of dynamic networks, IEEE Transactions on Neural Networks 18 (2007) 14–27. https://doi.org/10.1109/TNN.2006.882371.
[220] G.A. Ridlehoover, R.C. Seagrave, Optimization of Van de Vusse Reaction Kinetics Using Semibatch Reactor Operation, Industrial and Engineering Chemistry Fundamentals 12 (1973) 444–447. https://doi.org/10.1021/i160048a008.
[221] H. Ojeda-elizarras, R. Maya-yescas, S. Hern, J. Gabriel, S. Hern, A.J. Castro-montoya, Fuzzy Control of a Non-Linear System With Inverse Response : Van De Vusse, 2 (2013) 1–5.
[222] A.S. Chinen, J.C. Morgan, B. Omell, D. Bhattacharyya, D.C. Miller, Dynamic Data Reconciliation and Validation of a Dynamic Model for Solvent-Based CO 2 Capture Using Pilot-Plant Data, Industrial and Engineering Chemistry Research 58 (2019) 1978–1993. https://doi.org/10.1021/acs.iecr.8b04489.
[223] V. Topuz, Hourly Traffic Flow Prediction Using Different ANN Models, Urban Transport and Hybrid Vehicles (2010). https://doi.org/10.5772/10177.
[224] Z. Boussaada, O. Curea, A. Remaci, H. Camblong, N.M. Bellaaj, A nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation, Energies 11 (2018). https://doi.org/10.3390/en11030620.

[225] H.T. Siegelmann, B.G. Horne, C.L. Giles, Computational capabilities of recurrent NARX neural networks, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 27 (1997) 208–215. https://doi.org/10.1109/3477.558801.
[226] L.T. Biegler, Nonlinear Programming, Society for Industrial and Applied Mathematics, 2010. https://doi.org/10.1137/1.9780898719383.
[227] C.M. Hurvich, C.L. Tsai, Regression and time series model selection in small samples, Biometrika 76 (1989) 297–307. https://doi.org/10.1093/biomet/76.2.297.
[228] M.H. van Emden, An Analysis of Complexity, Mathematisch Centrum, 1971.
[229] P. Akula, J. Eslick, D. Bhattacharyya, D.C. Miller, Model Development, Validation, and Optimization of an MEA-Based Post-Combustion CO2Capture Process under Part-Load and Variable Capture Operations, Industrial and Engineering Chemistry Research 60 (2021) 5176–5193. https://doi.org/10.1021/acs.iecr.0c05035.
[230] J.C. Morgan, A.S. Chinen, C. Anderson-Cook, C. Tong, J. Carroll, C. Saha, B. Omell, D. Bhattacharyya, M. Matuszewski, K.S. Bhat, D.C. Miller, Development of a framework for sequential Bayesian design of experiments: Application to a pilot-scale solvent-based CO2 capture process, Applied Energy 262 (2020) 114533. https://doi.org/10.1016/j.apenergy.2020.114533.
[231] J.C. Morgan, A. Soares Chinen, B. Omell, D. Bhattacharyya, C. Tong, D.C. Miller, B. Buschle, M. Lucquiaud, Development of a Rigorous Modeling Framework for Solvent-Based CO2 Capture. Part 2: Steady-State Validation and Uncertainty Quantification with Pilot Plant Data, Industrial and Engineering Chemistry Research 57 (2018) 10464–10481. https://doi.org/10.1021/acs.iecr.8b01472.
[232] A. Soares Chinen, J.C. Morgan, B. Omell, D. Bhattacharyya, C. Tong, D.C. Miller, Development of a Rigorous Modeling Framework for Solvent-Based CO2 Capture. 1. Hydraulic and Mass Transfer Models and Their Uncertainty Quantification, Industrial and Engineering Chemistry Research 57 (2018) 10448–10463. https://doi.org/10.1021/acs.iecr.8b01471.
[233] A. Mukherjee, S. Adeyemo, D. Bhattacharyya, All-nonlinear static-dynamic neural networks versus Bayesian machine learning for data-driven modelling of chemical processes, The Canadian Journal of Chemical Engineering (2024). https://doi.org/10.1002/cjce.25379.
[234] S. Adeyemo, D. Bhattacharyya, Optimal nonlinear dynamic sparse model selection and Bayesian parameter estimation for nonlinear systems, Computers and Chemical Engineering 180 (2024) 108502. https://doi.org/10.1016/j.compchemeng.2023.108502.
[235] J. Sansana, M.N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L.H. Chiang, M.S. Reis, Recent trends on hybrid modeling for Industry 4.0, Computers and Chemical Engineering 151 (2021) 107365. https://doi.org/10.1016/j.compchemeng.2021.107365.
[236] H. Zheng, Z. Huang, G. Lin, PCNN: A physics-constrained neural network for multiphase flows, (2021) 1–21.
[237] A. Kumar, S. Ridha, M. Narahari, S.U. Ilyas, Physics-guided deep neural network to characterize non-Newtonian fluid flow for optimal use of energy resources, Expert Systems with Applications 183 (2021) 115409. https://doi.org/10.1016/j.eswa.2021.115409.
[238] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: a review, Acta Mechanica Sinica 37 (2021) 1727–1738. https://doi.org/10.1007/s10409-021-01148-1.
[239] A. Carranza-Abaid, J.P. Jakobsen, Neural network programming: Integrating first principles into machine learning models, Computers & Chemical Engineering 163 (2022) 107858. https://doi.org/10.1016/j.compchemeng.2022.107858.

[240] T.A. Ihunde, O. Olorode, Application of physics informed neural networks to compositional modeling, Journal of Petroleum Science and Engineering 211 (2022) 110175. https://doi.org/10.1016/j.petrol.2022.110175.
[241] R. Laubscher, P. Rousseau, Application of a mixed variable physics-informed neural network to solve the incompressible steady-state and transient mass, momentum, and energy conservation equations for flow over in-line heated tubes, Applied Soft Computing 114 (2022) 108050. https://doi.org/10.1016/j.asoc.2021.108050.
[242] Y. Zhu, N. Zabaras, P.S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, Journal of Computational Physics 394 (2019) 56–81. https://doi.org/10.1016/j.jcp.2019.05.024.
[243] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Reviews Physics 3 (2021) 422–440. https://doi.org/10.1038/s42254-021-00314-5.
[244] T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, P. Gentine, Enforcing Analytic Constraints in Neural Networks Emulating Physical Systems, Physical Review Letters 126 (2021). https://doi.org/10.1103/PhysRevLett.126.098302.
[245] H. Chen, G.E.C. Flores, C. Li, Physics-informed neural networks with hard linear equality constraints, Computers & Chemical Engineering 189 (2024) 108764. https://doi.org/10.1016/j.compchemeng.2024.108764.
[246] J. Stiasny, G.S. Misyris, S. Chatzivasileiadis, Transient Stability Analysis with Physics-Informed Neural Networks, (2021).
[247] L.B. de Giuli, A. La Bella, R. Scattolini, Physics-Informed Neural Network Modeling and Predictive Control of District Heating Systems, IEEE Transactions on Control Systems Technology 32 (2024) 1182–1195. https://doi.org/10.1109/TCST.2024.3355476.
[248] C. Hua, X. Guan, Output Feedback Stabilization for Time-Delay Nonlinear Interconnected Systems Using Neural Networks, IEEE Transactions on Neural Networks 19 (2008) 673–688. https://doi.org/10.1109/TNN.2007.912318.
[249] B. Huang, J. Wang, Applications of Physics-Informed Neural Networks in Power Systems - A Review, IEEE Transactions on Power Systems 38 (2023) 572–588. https://doi.org/10.1109/TPWRS.2022.3162473.
[250] L.D. González, V.M. Zavala, New paradigms for exploiting parallel experiments in Bayesian optimization, Computers & Chemical Engineering 170 (2023) 108110. https://doi.org/10.1016/j.compchemeng.2022.108110.
[251] B. Sun, C. Yang, Y. Wang, W. Gui, I. Craig, L. Olivier, A comprehensive hybrid first principles/machine learning modeling framework for complex industrial processes, Journal of Process Control 86 (2020) 30–43. https://doi.org/10.1016/j.jprocont.2019.11.012.
[252] I.T. Cameron, S. Engell, C. Georgakis, N. Asprion, D. Bonvin, F. Gao, D.I. Gerogiorgis, I.E. Grossmann, S. Macchietto, H.A. Preisig, B.R. Young, Education in Process Systems Engineering: Why it matters more than ever and how it can be structured, Computers & Chemical Engineering 126 (2019) 102–112. https://doi.org/10.1016/j.compchemeng.2019.03.036.
[253] Y. Kishimoto, D.S. Bernstein, Thermodynamic modelling of interconnected systems Part I: conservative coupling, Journal of Sound and Vibration 182 (1995) 23–58. https://doi.org/10.1006/jsvi.1995.0181.
[254] M. El Fallaki Idrissi, F. Praud, F. Meraghni, F. Chinesta, G. Chatzigeorgiou, Multiscale Thermodynamics-Informed Neural Networks (MuTINN) towards fast and frugal inelastic

computation of woven composite structures, Journal of the Mechanics and Physics of Solids 186 (2024) 105604. https://doi.org/10.1016/j.jmps.2024.105604.
[255] Z. Wu, H. Wang, C. He, B. Zhang, T. Xu, Q. Chen, The Application of Physics-Informed Machine Learning in Multiphysics Modeling in Chemical Engineering, Industrial & Engineering Chemistry Research 62 (2023) 18178–18204. https://doi.org/10.1021/acs.iecr.3c02383.
[256] A. Parente, N. Swaminathan, Data-driven models and digital twins for sustainable combustion technologies, iScience 27 (2024) 109349. https://doi.org/10.1016/j.isci.2024.109349.
[257] J. Drgoňa, A. Tuor, D. Vrabie, Learning Constrained Parametric Differentiable Predictive Control Policies With Guarantees, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2024) 1–12. https://doi.org/10.1109/TSMC.2024.3368026.
[258] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Programming 106 (2006) 25–57. https://doi.org/10.1007/s10107-004-0559-y.
[259] W.E. Hart, C.D. Laird, J.-P. Watson, D.L. Woodruff, G.A. Hackebeil, B.L. Nicholson, J.D. Siirola, Pyomo — Optimization Modeling in Python, Springer International Publishing, Cham, 2017. https://doi.org/10.1007/978-3-319-58821-6.
[260] D. Beahr, V. Saini, D. Bhattacharyya, S. Seachman, C. Boohaker, Estimation-based model predictive control with objective prioritization for mutually exclusive objectives: Application to a power plant, Journal of Process Control 141 (2024) 103268. https://doi.org/10.1016/j.jprocont.2024.103268.
[261] G.V. Jeffreys, A Problem in Chemical Engineering Design: The Manufacture of Acetic Anhydride, Institution of Chemical Engineers, 1964.
[262] G. Froment, H. Pijcke, G. Goethals, Thermal cracking of acetone—I. Influence of the process variables on the product distribution, Chemical Engineering Science 13 (1961) 173–179. https://doi.org/10.1016/0009-2509(61)80008-0.
[263] G. Froment, H. Pijcke, G. Goethals, Thermal cracking of acetone—II. Deriving rate-equations from non-isothermal flow experiments, Chemical Engineering Science 13 (1961) 180–189. https://doi.org/10.1016/0009-2509(61)80009-2.
[264] A. Mukherjee, D. Bhattacharyya, Development of Mass, Energy, and Thermodynamics Constrained Steady-State and Dynamic Neural Networks for Interconnected Chemical Systems, Chemical Engineering Science (2024) Under Review.
[265] H.S. Fogler, Elements of Chemical Reaction Engineering, Fourth Edi, Prentice Hall, 2017.
[266] J.R. Elliott, C.T. Lira, Introductory chemical engineering thermodynamics, Upper Saddle River, NJ : Prentice Hall PTR, [1999] ©1999, 1999.
[267] H.C. Longuet-Higgins, The statistical thermodynamics of multicomponent systems, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 205 (1951) 247–269.
[268] P. Akula, A. Lee, J. Eslick, D. Bhattacharyya, D.C. Miller, A modified electrolyte non-random two-liquid model with analytical expression for excess enthalpy: Application to the MEA-H2O-CO2 system, AIChE Journal 69 (2023). https://doi.org/10.1002/aic.17935.
[269] C. Chen, Y. Song, Generalized electrolyte-NRTL model for mixed-solvent electrolyte systems, AIChE Journal 50 (2004) 1928–1941. https://doi.org/10.1002/aic.10151.
[270] P. Paul, D. Bhattacharyya, R. Turton, S.E. Zitney, Sensor network design for maximizing process efficiency: An algorithm and its application, AIChE Journal 61 (2015) 464–476. https://doi.org/10.1002/aic.14649.
[271] Q. Huang, D. Bhattacharyya, Optimal sensor network design for multi-scale, time-varying differential algebraic equation systems: Application to an entrained-flow gasifier refractory brick,

Computers and Chemical Engineering 141 (2020) 106985. https://doi.org/10.1016/j.compchemeng.2020.106985.
[272] P. Boniecki, M. Zaborowicz, A. Sujak, Comparison of MLP and RBF neural models on the example graphical classification, in: X. Jiang, H. Fujita (Eds.), Thirteenth International Conference on Digital Image Processing (ICDIP 2021), SPIE, 2021: p. 40. https://doi.org/10.1117/12.2600796.
[273] M. Motahari-Nezhad, S.M. Jafari, Comparison of MLP and RBF neural networks for bearing remaining useful life prediction based on acoustic emission, Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology 237 (2023) 129–148. https://doi.org/10.1177/13506501221106556.
[274] A.R. Senthil Kumar, K.P. Sudheer, S.K. Jain, P.K. Agarwal, Rainfall-runoff modelling using artificial neural networks: Comparison of network types, Hydrological Processes 19 (2005) 1277–1291. https://doi.org/10.1002/hyp.5581.
[275] T. Xie, H. Yu, B. Wilamowski, Comparison between traditional neural networks and radial basis function networks, Proceedings - ISIE 2011: 2011 IEEE International Symposium on Industrial Electronics (2011) 1194–1199. https://doi.org/10.1109/ISIE.2011.5984328.
[276] M. Zare, H.R. Pourghasemi, M. Vafakhah, B. Pradhan, Landslide susceptibility mapping at Vaz Watershed (Iran) using an artificial neural network model: A comparison between multilayer perceptron (MLP) and radial basic function (RBF) algorithms, Arabian Journal of Geosciences 6 (2013) 2873–2888. https://doi.org/10.1007/s12517-012-0610-x.
[277] F. Cheshmberah, H. Fathizad, G.A. Parad, S. Shojaeifar, Comparison of RBF and MLP neural network performance and regression analysis to estimate carbon sequestration, International Journal of Environmental Science and Technology 17 (2020) 3891–3900. https://doi.org/10.1007/s13762-020-02696-y.
[278] H. Leung, T. Lo, S. Wang, Prediction of noisy chaotic time series using an optimal radial basis function neural network, IEEE Transactions on Neural Networks 12 (2001) 1163–1172. https://doi.org/10.1109/72.950144.
[279] C. Panchapakesan, M. Palaniswami, D. Ralph, C. Manzie, Effects of moving the centers in an RBF network, IEEE Transactions on Neural Networks 13 (2002) 1299–1307. https://doi.org/10.1109/TNN.2002.804286.
[280] N.J. Wu, Predicting the compressive strength of concrete using an rbf-ann model, Applied Sciences (Switzerland) 11 (2021). https://doi.org/10.3390/app11146382.
[281] G.A. Montazer, D. Giveki, M. Karami, H. Rastegar, Radial basis function neural networks: A review, Comput. Rev. J 1 (2018) 52–74.
[282] D. Du, K. Li, M. Fei, A fast multi-output RBF neural network construction method, Neurocomputing 73 (2010) 2196–2202. https://doi.org/10.1016/j.neucom.2010.01.014.
[283] S. V. Chakravarthy, J. Ghosh, Scale-based clustering using the radial basis function network, IEEE Transactions on Neural Networks 7 (1996) 1250–1261. https://doi.org/10.1109/72.536318.
[284] A. Alexandridis, E. Chondrodima, H. Sarimveis, Radial basis function network training using a nonsymmetric partition of the input space and particle swarm optimization, IEEE Transactions on Neural Networks and Learning Systems 24 (2013) 219–230. https://doi.org/10.1109/TNNLS.2012.2227794.
[285] M.Y. Mashor, Hybrid Training Algorithm for RBF Network, International Journal of the Computer, The Internet and Management 8 (2000) 50–65.
[286] A.D. Niros, G.E. Tsekouras, A novel training algorithm for RBF neural network using a hybrid fuzzy clustering approach, Fuzzy Sets and Systems 193 (2012) 62–84. https://doi.org/10.1016/j.fss.2011.08.011.

[287] A. Gosain, S. Dahiya, Performance Analysis of Various Fuzzy Clustering Algorithms: A Review, Procedia Computer Science 79 (2016) 100–111. https://doi.org/10.1016/j.procs.2016.03.014.
[288] Y. Zhang, C. Gong, H. Fang, H. Su, C. Li, A. Da Ronch, An efficient space division–based width optimization method for RBF network using fuzzy clustering algorithms, Structural and Multidisciplinary Optimization 60 (2019) 461–480. https://doi.org/10.1007/s00158-019-02217-7.
[289] M. Foqaha, Hybrid Approach to Optimize the Centers of Radial Basis Function Neural Network Using Particle Swarm Optimization, Journal of Computers 12 (2017) 396–407. https://doi.org/10.17706/jcp.12.5.396-407.
[290] Z.Q. Zhao, X.D. Wu, C.Y. Lu, H. Glotin, J. Gao, Optimizing widths with PSO for center selection of Gaussian radial basis function networks, Science China Information Sciences 57 (2014) 1–17. https://doi.org/10.1007/s11432-013-4850-5.
[291] Q. Wang, L. Pan, K.Y. Lee, Z. Wu, Deep-learning modeling and control optimization framework for intelligent thermal power plants: A practice on superheated steam temperature, Korean Journal of Chemical Engineering 38 (2021) 1983–2002. https://doi.org/10.1007/s11814-021-0865-6.
[292] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Transactions on Neural Networks 9 (1998) 987–1000. https://doi.org/10.1109/72.712178.
[293] H. Liu, J. He, The application of dynamic K-means clustering algorithm in the center selection of RBF neural networks, 3rd International Conference on Genetic and Evolutionary Computing, WGEC 2009 (2009) 488–491. https://doi.org/10.1109/WGEC.2009.112.
[294] K.P. Sinaga, M.S. Yang, Unsupervised K-means clustering algorithm, IEEE Access 8 (2020) 80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796.
[295] M. Awad, H. Pomares, I. Rojas, O. Salameh, M. Hamdon, Prediction of time series using RBF neural networks: A new approach of clustering, International Arab Journal of Information Technology 6 (2009) 138–143.
[296] A. Nagpal, A. Jatain, D. Gaur, Review based on data clustering algorithms, 2013 IEEE Conference on Information and Communication Technologies, ICT 2013 (2013) 298–303. https://doi.org/10.1109/CICT.2013.6558109.
[297] J. Raitoharju, S. Kiranyaz, M. Gabbouj, Training Radial Basis Function Neural Networks for Classification via Class-Specific Clustering, IEEE Transactions on Neural Networks and Learning Systems 27 (2016) 2458–2471. https://doi.org/10.1109/TNNLS.2015.2497286.
[298] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O.P. Patel, A. Tiwari, M.J. Er, W. Ding, C.T. Lin, A review of clustering techniques and developments, Neurocomputing 267 (2017) 664–681. https://doi.org/10.1016/j.neucom.2017.06.053.
[299] M. Mittal, L.M. Goyal, D.J. Hemanth, J.K. Sethi, Clustering approaches for high-dimensional databases: A review, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9 (2019) 1–14. https://doi.org/10.1002/widm.1300.
[300] S.L. Tilahun, J.M.T. Ngnotchouye, N.N. Hamadneh, Continuous versions of firefly algorithm: a review, Artificial Intelligence Review 51 (2019) 445–492. https://doi.org/10.1007/s10462-017-9568-0.
[301] S.L. Tilahun, H.C. Ong, Prey-predator algorithm: A new metaheuristic algorithm for optimization problems, International Journal of Information Technology and Decision Making 14 (2015) 1331–1352. https://doi.org/10.1142/S021962201450031X.
[302] R. Neruda, P. Kudová, Learning methods for radial basis function networks, Future Generation Computer Systems 21 (2005) 1131–1142. https://doi.org/10.1016/j.future.2004.03.013.

[303] K. Tyagi, C. Rane, B. Irie, M. Manry, Multistage Newton's Approach for Training Radial Basis Function Neural Networks, SN Computer Science 2 (2021). https://doi.org/10.1007/s42979-021-00757-8.
[304] A. Alexandridis, E. Chondrodima, H. Sarimveis, Cooperative learning for radial basis function networks using particle swarm optimization, Elsevier B.V., 2016. https://doi.org/10.1016/j.asoc.2016.08.032.
[305] A. Alexandridis, E. Chondrodima, N. Giannopoulos, H. Sarimveis, A fast and efficient method for training categorical radial basis function networks, IEEE Transactions on Neural Networks and Learning Systems 28 (2017) 2831–2836. https://doi.org/10.1109/TNNLS.2016.2598722.
[306] C. Harpham, C.W. Dawson, M.R. Brown, A review of genetic algorithms applied to training radial basis function networks, Neural Computing and Applications 13 (2004) 193–201. https://doi.org/10.1007/s00521-004-0404-5.
[307] H. Bhavsar, M.H. Panchal, A Review on Support Vector Machine for Data Classification, International Journal of Advanced Research in Computer Engineering & Technology 1 (2012) 2278–1323.
[308] X. Hong, R.J. Mitchell, S. Chen, C.J. Harris, K. Li, G.W. Irwin, Model selection approaches for non-linear system identification: A review, International Journal of Systems Science 39 (2008) 925–946. https://doi.org/10.1080/00207720802083018.
[309] A. Hagg, M. Mensing, A. Asteroth, Evolving parsimonious networks by mixing activation functions, GECCO 2017 - Proceedings of the 2017 Genetic and Evolutionary Computation Conference (2017) 425–432. https://doi.org/10.1145/3071178.3071275.
[310] X. Zou, Z. Wang, Q. Li, W. Sheng, Integration of residual network and convolutional neural network along with various activation functions and global pooling for time series classification, Neurocomputing 367 (2019) 39–45. https://doi.org/10.1016/j.neucom.2019.08.023.
[311] K.-L. Li, C.-J. Zhai, J.-M. Xu, Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN, in: 2017 Chinese Automation Congress (CAC), 2017: pp. 2804–2807. https://doi.org/10.1109/CAC.2017.8243253.
[312] P. Pecháč, M. Ságа, Memetic Algorithm with Normalized RBF ANN for Approximation of Objective Function and Secondary RBF ANN for Error Mapping, Procedia Engineering 177 (2017) 540–547. https://doi.org/10.1016/j.proeng.2017.02.258.
[313] E. Ghanbari-Adivi, M. Ehteram, A. Farrokhi, Z. Sheikh Khozani, Combining Radial Basis Function Neural Network Models and Inclusive Multiple Models for Predicting Suspended Sediment Loads, Water Resources Management 36 (2022) 4313–4342. https://doi.org/10.1007/s11269-022-03256-4.
[314] B. Doty, S. Mihalas, A. Arkhipov, A. Piet, Heterogeneous {\textquoteleft}cell types{\textquoteright} can improve performance of deep neural networks, bioRxiv (2021). https://doi.org/10.1101/2021.06.21.449346.
[315] S. Li, Q. Chen, G.-B. Huang, Dynamic temperature modeling of continuous annealing furnace using GGAP-RBF neural network, Neurocomputing 69 (2006) 523–536. https://doi.org/10.1016/j.neucom.2005.01.008.
[316] Ch. Venkateswarlu, K. Venkat Rao, Dynamic recurrent radial basis function network model predictive control of unstable nonlinear processes, Chemical Engineering Science 60 (2005) 6718–6732. https://doi.org/10.1016/j.ces.2005.03.070.
[317] T. Liu, S. Chen, S. Liang, D. Du, C.J. Harris, Fast tunable gradient RBF networks for online modeling of nonlinear and nonstationary dynamic processes, Journal of Process Control 93 (2020) 53–65. https://doi.org/10.1016/j.jprocont.2020.07.009.

[318] G. Su, L. Peng, L. Hu, A Gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis, Structural Safety 68 (2017) 97–109. https://doi.org/10.1016/j.strusafe.2017.06.003.
[319] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (1974) 716–723. https://doi.org/10.1109/TAC.1974.1100705.
[320] A. Lee, J.H. Ghouse, J.C. Eslick, C.D. Laird, J.D. Siirola, M.A. Zamarripa, D. Gunter, J.H. Shinn, A.W. Dowling, D. Bhattacharyya, L.T. Biegler, A.P. Burgard, D.C. Miller, The IDAES process modeling framework and model library—Flexibility for process simulation and optimization, Journal of Advanced Manufacturing and Processing 3 (2021). https://doi.org/10.1002/amp2.10095.
[321] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, (2014). https://doi.org/10.48550/arXiv.1412.6980.
[322] C.P. Tomazella, M.S. Nagano, A comprehensive review of Branch-and-Bound algorithms: Guidelines and directions for further research on the flowshop scheduling problem, Expert Systems with Applications 158 (2020) 113556. https://doi.org/10.1016/j.eswa.2020.113556.
[323] J. Bradbury, R. Frostig, P. Hawkins, M.J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: Composable transformations of Python+Numpy programs, (2018). http://github.com/google/jax.
[324] M. Lin, Automatic Functional Differentiation in JAX, in: The Twelfth International Conference on Learning Representations (ICLR 2024), 2023.
[325] M. Zhang, Y. Zhang, Z. Gao, X. He, An Improved DDPG and Its Application Based on the Double-Layer BP Neural Network, IEEE Access 8 (2020) 177734–177744. https://doi.org/10.1109/ACCESS.2020.3020590.
[326] I.D. Coope, C.J. Price, A Modified BFGS Formula Maintaining Positive Definiteness with Armijo-Goldstein Steplengths, Journal of Computational Mathematics 13 (1995) 156–160.
[327] A. Neumaier, M. Kimiaei, An improvement of the Goldstein line search, Optimization Letters (2024). https://doi.org/10.1007/s11590-024-02110-3.
[328] E. Frank, Fully supervised training of Gaussian radial basis function networks in WEKA, Department of Computer Science, University of Waikato, Tech. Rep 04 (2014) 4.
[329] A.O. Hoori, Y. Motai, Multicolumn RBF Network, IEEE Transactions on Neural Networks and Learning Systems 29 (2018) 766–778. https://doi.org/10.1109/TNNLS.2017.2650865.
[330] S.A. Dyer, J.S. Dyer, Cubic-spline interpolation. 1, IEEE Instrumentation & Measurement Magazine 4 (2001) 44–46. https://doi.org/10.1109/5289.911175.
[331] J.A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, M. Diehl, CasADi: a software framework for nonlinear optimization and optimal control, Mathematical Programming Computation 11 (2019) 1–36. https://doi.org/10.1007/s12532-018-0139-4.
[332] J.D. Kelly, J.D. Hedengren, A steady-state detection (SSD) algorithm to detect non-stationary drifts in processes, Journal of Process Control 23 (2013) 326–331. https://doi.org/10.1016/j.jprocont.2012.12.001.
[333] S. Cao, R.R. Rhinehart, An efficient method for on-line identification of steady state, Journal of Process Control 5 (1995) 363–374. https://doi.org/10.1016/0959-1524(95)00009-F.
[334] S. Cao, R.R. Rhinehart, Critical values for a steady-state identifier, Journal of Process Control 7 (1997) 149–152. https://doi.org/10.1016/S0959-1524(96)00026-1.
[335] R.R. Rhinehart, Automated steady and transient state identification in noisy processes, in: 2013 American Control Conference, IEEE, 2013: pp. 4477–4493. https://doi.org/10.1109/ACC.2013.6580530.

[336] G. Zaccone, M.R. Karim, Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python, 2nd Edition, Packt Publishing, Birmingham, United Kingdom, 2018.
[337] V. Saini, A. Mukherjee, S. Adeyemo, D. Bhattacharyya, Grey-Box Modeling and Estimation for Health Monitoring of High Temperature Boiler Components, in: 2023 AIChE Annual Meeting, AIChE, 2023. https://aiche.confex.com/aiche/2023/prelim.cgi/Paper/670487.
[338] M. Granda, M. Trojan, D. Taler, CFD analysis of steam superheater operation in steady and transient state, Energy 199 (2020) 117423. https://doi.org/10.1016/j.energy.2020.117423.
[339] B. Sohlberg, E.W. Jacobsen, GREY BOX MODELLING – BRANCHES AND EXPERIENCES, IFAC Proceedings Volumes 41 (2008) 11415–11420. https://doi.org/10.3182/20080706-5-KR-1001.01934.
[340] A. Mukherjee, S. Adeyemo, D. Bhattacharyya, All-nonlinear static-dynamic neural networks versus Bayesian machine learning for data-driven modelling of chemical processes, The Canadian Journal of Chemical Engineering (2024). https://doi.org/10.1002/cjce.25379.
[341] M. Mohamadi-Baghmolaei, P. Zahedizadeh, M. Khajeh, S. Zendehboudi, N.M.C. Saady, Developing a hybridized thermodynamic and data-driven model for catalytic supercritical water gasification of biomass for hydrogen production, Energy Conversion and Management 307 (2024) 118302. https://doi.org/10.1016/j.enconman.2024.118302.
[342] W. Chen, Y. Liang, X. Luo, J. Chen, Z. Yang, Y. Chen, Artificial neural network grey-box model for design and optimization of 50 MWe-scale combined supercritical CO2 Brayton cycle-ORC coal-fired power plant, Energy Conversion and Management 249 (2021) 114821. https://doi.org/10.1016/j.enconman.2021.114821.
[343] D.C. Psichogios, L.H. Ungar, A hybrid neural network-first principles approach to process modeling, AIChE Journal 38 (1992) 1499–1511. https://doi.org/10.1002/aic.690381003.
[344] M.S.F. Bangi, J.S.I. Kwon, Deep hybrid modeling of chemical process: Application to hydraulic fracturing, Computers and Chemical Engineering 134 (2020). https://doi.org/10.1016/j.compchemeng.2019.106696.
[345] L. Rajulapati, S. Chinta, B. Shyamala, R. Rengaswamy, Integration of machine learning and first principles models, AIChE Journal 68 (2022) 1–17. https://doi.org/10.1002/aic.17715.
[346] D. Machalek, J. Tuttle, K. Andersson, K.M. Powell, Dynamic energy system modeling using hybrid physics-based and machine learning encoder–decoder models, Energy and AI 9 (2022) 100172. https://doi.org/10.1016/j.egyai.2022.100172.
[347] D. Lee, A. Jayaraman, J.S. Kwon, Development of a hybrid model for a partially known intracellular signaling pathway through correction term estimation and neural network modeling, 2020. https://doi.org/10.1371/journal.pcbi.1008472.
[348] M.S.F. Bangi, K. Kao, J.S.-I. Kwon, Physics-informed neural networks for hybrid modeling of lab-scale batch fermentation for β-carotene production using Saccharomyces cerevisiae, Chemical Engineering Research and Design 179 (2022) 415–423. https://doi.org/10.1016/j.cherd.2022.01.041.
[349] A. Mukherjee, D. Bhattacharyya, Development of Steady-State and Dynamic Mass and Energy Constrained Neural Networks for Distributed Chemical Systems Using Noisy Transient Data, Industrial & Engineering Chemistry Research 63 (2024) 14211–14239. https://doi.org/10.1021/acs.iecr.4c01429.
[350] J. Yu, N. Petersen, P. Liu, Z. Li, M. Wirsum, Hybrid modelling and simulation of thermal systems of in-service power plants for digital twin development, Energy 260 (2022) 125088. https://doi.org/10.1016/j.energy.2022.125088.

[351] Z. Cui, J. Xu, W. Liu, G. Zhao, S. Ma, Data-driven modeling-based digital twin of supercritical coal-fired boiler for metal temperature anomaly detection, Energy 278 (2023) 127959. https://doi.org/10.1016/j.energy.2023.127959.
[352] J.P. Spinti, P.J. Smith, S.T. Smith, Atikokan Digital Twin: Machine learning in a biomass energy system, Applied Energy 310 (2022) 118436. https://doi.org/10.1016/j.apenergy.2021.118436.
[353] W. Bradley, J. Kim, Z. Kilwein, L. Blakely, M. Eydenberg, J. Jalvin, C. Laird, F. Boukouvala, Perspectives on the integration between first-principles and data-driven modeling, Computers and Chemical Engineering 166 (2022) 107898. https://doi.org/10.1016/j.compchemeng.2022.107898.
[354] A. Chakraborty, S. Serneels, H. Claussen, V. Venkatasubramanian, Hybrid AI Models in Chemical Engineering – A Purpose-driven Perspective, in: 2022: pp. 1507–1512. https://doi.org/10.1016/B978-0-323-95879-0.50252-6.
[355] S. Zendehboudi, N. Rezaei, A. Lohi, Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review, Applied Energy 228 (2018) 2539–2566. https://doi.org/10.1016/j.apenergy.2018.06.051.
[356] A. Mukherjee, V. Saini, S. Adeyemo, D. Bhattacharyya, Development of Hybrid First Principles–Artificial Intelligence Models: Applications to an Industrial Steam Superheater System, in: 2023 AIChE Annual Meeting, AIChE, 2023. https://aiche.confex.com/aiche/2023/meetingapp.cgi/Paper/668352.
[357] Y. Chen, M. Ierapetritou, A framework of hybrid model development with identification of plant-model mismatch, AIChE Journal 66 (2020) 1–16. https://doi.org/10.1002/aic.16996.
[358] D.S. Lee, P.A. Vanrolleghem, M.P. Jong, Parallel hybrid modeling methods for a full-scale cokes wastewater treatment plant, Journal of Biotechnology 115 (2005) 317–328. https://doi.org/10.1016/j.jbiotec.2004.09.001.
[359] A. Ostace, Y.-Y. Chen, R. Parker, D.S. Mebane, C.O. Okoli, A. Lee, A. Tong, L.-S. Fan, L.T. Biegler, A.P. Burgard, D.C. Miller, D. Bhattacharyya, Kinetic model development and Bayesian uncertainty quantification for the complete reduction of Fe-based oxygen carriers with CH4, CO, and H2 for chemical looping combustion, Chemical Engineering Science 252 (2022) 117512. https://doi.org/10.1016/j.ces.2022.117512.
[360] M. von Stosch, R. Oliveira, J. Peres, S. Feyo de Azevedo, Hybrid semi-parametric modeling in process systems engineering: Past, present and future, Computers & Chemical Engineering 60 (2014) 86–101. https://doi.org/10.1016/j.compchemeng.2013.08.008.
[361] L.T. Biegler, Nonlinear Programming, Society for Industrial and Applied Mathematics, 2010. https://doi.org/10.1137/1.9780898719383.
[362] J. Cochran, M. Miller, O. Zinaman, M. Milligan, D. Arent, B. Palmintier, M. O'Malley, S. Mueller, E. Lannoye, A. Tuohy, B. Kujala, M. Sommer, H. Holttinen, J. Kiviluoma, S.K. Soonee, Flexibility in 21st Century Power Systems, National Renewable Energy Laboratory (NREL), Golden, CO (United States), 2014. https://doi.org/10.2172/1130630.
[363] N. Kumar, P. Besuner, S. Lefton, D. Agan, D. Hilleman, Power Plant Cycling Costs, National Renewable Energy Laboratory (NREL), Golden, CO (United States), 2012. https://doi.org/10.2172/1046269.
[364] Y. Wang, D. Bhattacharyya, R. Turton, Evaluation of Novel Configurations of Natural Gas Combined Cycle (NGCC) Power Plants for Load-Following Operation using Dynamic Modeling and Optimization, Energy Fuels 34 (2020) 1053–1070. https://doi.org/10.1021/acs.energyfuels.9b03036.

[365] Q.M. Le, J. Ma, D. Bhattacharyya, S.E. Zitney, A.P. Burgard, Design and Multiobjective Dynamic Optimization of Superheaters for Load-Following Operation in Pulverized Coal Power Plants, Ind. Eng. Chem. Res. 63 (2024) 330–344. https://doi.org/10.1021/acs.iecr.3c02130.
[366] V. Saini, D. Bhattacharyya, State and Parameter Estimation of Complex Dynamic Systems Using Nonlinear Grey Box Modeling: Application to an Industrial Steam Superheater System, in: 2022 AIChE Annual Meeting, AIChE, 2022. https://aiche.confex.com/aiche/2022/prelim.cgi/Paper/648054.
[367] J. Qi, K. Zhou, J. Huang, X. Si, Influence of temperature on the oxide spallation of T91 alloy superheater tubes in power plant, Applied Thermal Engineering 128 (2018) 244–252. https://doi.org/10.1016/j.applthermaleng.2017.08.139.
[368] A.S. Sabau, I.G. Wright, J.P. Shingledecker, Oxide scale exfoliation and regrowth in TP347H superheater tubes, Materials and Corrosion 63 (2012) 896–908. https://doi.org/10.1002/maco.201206640.
[369] L.J. Mendes N., E. Bazzo, J. Toste Azevedo, Thermal conductivity analysis of an ash deposit on boiler superheater, Powder Technology 318 (2017) 329–336. https://doi.org/10.1016/j.powtec.2017.06.005.
[370] J. Qi, K. Zhou, J. Huang, X. Si, Numerical simulation of the heat transfer of superheater tubes in power plants considering oxide scale, International Journal of Heat and Mass Transfer 122 (2018) 929–938. https://doi.org/10.1016/j.ijheatmasstransfer.2018.01.119.
[371] Y. Cai, K. Tay, Z. Zheng, W. Yang, H. Wang, G. Zeng, Z. Li, S. Keng Boon, P. Subbaiah, Modeling of ash formation and deposition processes in coal and biomass fired boilers: A comprehensive review, Applied Energy 230 (2018) 1447–1544. https://doi.org/10.1016/j.apenergy.2018.08.084.
[372] C.S. Somayajula, D. Bhattacharyya, X. Liu, S. Hu, Optimal Sensor Network Design for Maximizing Net Present Value and Its Application to Corrosion Monitoring in a Power Plant, in: 2023 AIChE Annual Meeting, AIChE, 2023. https://aiche.confex.com/aiche/2023/prelim.cgi/Paper/670868.
[373] C.S. Somayajula, State Estimation and Cost Optimal Sensor Network Design for Electrochemical Sensor Based Corrosion Monitoring, in: 2023 AIChE Annual Meeting, AIChE, 2023. https://aiche.confex.com/aiche/2023/meetingapp.cgi/Paper/674676.
[374] C.S. Somayajula, State Estimation and Economic Analysis for Electrochemical Sensor-Based Corrosion Monitoring, Graduate Theses, Dissertations, and Problem Reports (2024). https://doi.org/10.33915/etd.12571.
[375] A. Hamid, A.H. Hasan, S.N. Azhari, Z. Harun, Z.A. Putra, Hybrid modelling for remote process monitoring and optimisation, Digital Chemical Engineering 4 (2022) 100044. https://doi.org/10.1016/j.dche.2022.100044.
[376] L.M. Romeo, R. Gareta, Hybrid System for fouling control in biomass boilers, Engineering Applications of Artificial Intelligence 19 (2006) 915–925. https://doi.org/10.1016/j.engappai.2006.01.019.
[377] K. Megdouli, B. Tashtoush, P. Cinnella, Application of machine learning to enhance the performance of a two-stage, two-temperature ejector cycle driven by the waste heat of exhaust gas, Energy Conversion and Management 302 (2024) 118091. https://doi.org/10.1016/j.enconman.2024.118091.
[378] D. Santos, S. Costa, L. Rato, T. Gonçalves, I. Malico, P. Canhoto, F. Alvarez, M. Barão, A grey-box Neural Network Composite Model for an Industrial Heating Furnace, in: 25th Portuguese Conference on Pattern Recognition RECPAD2019, 2019: pp. 83–85.

[379] A. Chaibakhsh, Modelling and long-term simulation of a heat recovery steam generator, Mathematical and Computer Modelling of Dynamical Systems 19 (2013) 91–114. https://doi.org/10.1080/13873954.2012.698623.
[380] A. Aminzadeh, A.A. Safavi, A.R. Seifi, Development of a hybrid simulator of a fossil fuel steam power plant, in: 2003 European Control Conference (ECC), IEEE, Cambridge, UK, 2003: pp. 1905–1910. https://doi.org/10.23919/ECC.2003.7085244.
[381] H. Rusinowski, W. Stanek, Hybrid model of steam boiler, Energy 35 (2010) 1107–1113. https://doi.org/10.1016/j.energy.2009.06.004.
[382] T. Barszcz, P. Czop, Estimation of feedwater heater parameters based on a grey-box approach, International Journal of Applied Mathematics and Computer Science 21 (2011) 703–715. https://doi.org/10.2478/v10006-011-0056-4.
[383] T. Barszcz, P. Czop, J. Bednarz, Model-based evaluation of a power plant steam boiler system, Problemy Eksploatacji 2 (2011) 7–19.
[384] A.A. Farooq, A. Afram, N. Schulz, F. Janabi-Sharifi, Grey-box modeling of a low pressure electric boiler for domestic hot water system, Applied Thermal Engineering 84 (2015) 257–267. https://doi.org/10.1016/j.applthermaleng.2015.03.050.
[385] Y. Shi, T. Han, F. Cui, J. Wen, J. Jia, X. Pang, A hybrid prediction approach for enhancing heat transfer efficiency of coal-fired power plant boiler, Energy Reports 9 (2023) 658–668. https://doi.org/10.1016/j.egyr.2023.04.337.
[386] Y. Zhao, Y. Cai, H. Jiang, Recurrent Neural Network-Based Hybrid Modeling Method for Digital Twin of Boiler System in Coal-Fired Power Plant, Applied Sciences 13 (2023) 4905. https://doi.org/10.3390/app13084905.
[387] P. Wang, F. Si, Y. Cao, Z. Shao, S. Ren, Prediction of superheated steam temperature for thermal power plants using a novel integrated method based on the hybrid model and attention mechanism, Applied Thermal Engineering 203 (2022) 117899. https://doi.org/10.1016/j.applthermaleng.2021.117899.
[388] Q. Wang, L. Pan, K.Y. Lee, Z. Wu, Deep-learning modeling and control optimization framework for intelligent thermal power plants: A practice on superheated steam temperature, Korean J. Chem. Eng. 38 (2021) 1983–2002. https://doi.org/10.1007/s11814-021-0865-6.
[389] V. Saini, A. Mukherjee, S. Adeyemo, D. Bhattacharyya, Development of an Online Health Monitoring Framework for High Temperature Boiler Components By Using Hybrid First Principles-Data-Driven Models, in: 2024 AIChE Annual Meeting, AIChE, 2024. https://aiche.confex.com/aiche/2024/meetingapp.cgi/Paper/692107.
[390] P. Shah, M.Z. Sheriff, M.S.F. Bangi, C. Kravaris, J.S.I. Kwon, C. Botre, J. Hirota, Deep neural network-based hybrid modeling and experimental validation for an industry-scale fermentation process: Identification of time-varying dependencies among parameters, Chemical Engineering Journal 441 (2022) 135643. https://doi.org/10.1016/j.cej.2022.135643.
[391] N. Sitapure, J. Sang-Il Kwon, Introducing Hybrid Modeling with Time-Series-Transformers: A Comparative Study of Series and Parallel Approach in Batch Crystallization, Industrial & Engineering Chemistry Research (2023). https://doi.org/10.1021/acs.iecr.3c02624.
[392] K. Tewari, S. Shukla, R.K. Arya, Heat Exchanger Network Design Using MATLAB, Chem Eng & Technol 45 (2022) 508–516. https://doi.org/10.1002/ceat.202100543.
[393] L.F. Shampine, M.W. Reichelt, The MATLAB ODE Suite, SIAM Journal on Scientific Computing 18 (1997) 1–22. https://doi.org/10.1137/S1064827594276424.
[394] R.B. Dooley, Program on Technology Innovation : Oxide Growth and Exfoliation on Alloys Exposed to Steam, 2007.

[395] D. Taler, M. Trojan, J.M. Taler, Mathematical Modeling of Cross-Flow Tube Heat Exchangers With a Complex Flow Arrangement, Heat Transfer Engineering 35 (2014) 1334–1343. https://doi.org/10.1080/01457632.2013.876874.
[396] The Grade 22 Low Alloy Steel Handbook, EPRI, Palo Alto, CA 2006, 1012840.
[397] R. Purgert, J. Shingledecker, J. Pschirer, R. Ganta, P. Weitzel, J. Sarver, B. Vitalis, M. Gagliano, G. Stanko, P. Tortorelli, Boiler materials for ultra supercritical coal power plants, United States, 2015. https://doi.org/10.2172/1346714.
[398] Remaining Life Assessment of Austenitic Stainless Steel Superheater and Reheater Tubes, EPRI, Palo Alto, CA 2002, 1004517.
[399] P.R. Boyles, T. Fukui, M. Igarashi, A. Iseda, N. Komai, R. Masuyama, O. Miyahara, Y. Minami, H. Mimura, H. Okada, M. Prager, S. Yamamoto, Examination of Mechanical Properties and Corrosion of High Temperature Alloys after Long Term Service under Advanced Power Plant Boiler Conditions - The Eddystone Studies, Welding Research Council Bulletin,WRC 517 (2006).
[400] Remaining Life Assessment of SA213 T91 Steel Superheater and Reheater Tubes Subjected to Long-Term Overheat-Creep Damage and Potential Use Limitations: Glen Lyn Unit 6 Secondary Superheater Case Study, EPRI, Palo Alto, CA 2009, 1019328.
[401] F.P. Incropera, D.P. DeWitt, Fundamentals of Heat and Mass Transfer, 6th ed., Wiley, New York, 2007.
[402] Handbook of Oxide Scale Morphologies Formed in High-Temperature Steam, EPRI, Palo Alto, CA 2021., 3002018023.
[403] J. Lis, P. Kellard, Measurement of thermal conductivity of thin magnetite films, British J. Applied Physics 1 (n.d.) 1117–1123.
[404] Guidelines for Performing Probabilistic Analyses of Boiler Pressure Parts, EPRI, Palo Alto, CA 2000, 1000311.
[405] TULIP 2.0 – Tube Life Probability, EPRI, Palo Alto, CA 2006, 1010621.