

SANDIA REPORT

SAND2017-10178

Unlimited Release

Printed September 2017

Full State Feedback Control for Virtual Power Plants

A. P. Sakis Meliopoulos, George Cokkinides, Boqi Xie, Chiyang Zhong, Jay Johnson

Prepared by
Georgia Institute of Technology
Atlanta, Georgia

and

Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: 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, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent 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, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: <http://www.osti.gov/scitech>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: <http://www.ntis.gov/search>



SAND2017-10178
Unlimited Release
Printed September 2017

Full State Feedback Control for Virtual Power Plants

A P Sakis Meliopoulos, George Cokkinides, Boqi Xie, Chiyang Zhong
School of Electrical and Computer Engineering
Georgia Institute of Technology

Jay Johnson
Technical Monitor
Renewable and Distributed Systems Integration
Sandia National Laboratories

ACKNOWLEDGMENTS

The Virtual Power Plant project was supported by the Energy and Climate Laboratory Directed Research and Development Program at Sandia National Laboratories.

Contents

Section 1: Executive Summary.....	4
Section 2: Introduction	6
Section 3: Object-Oriented Device Modeling	12
Section 3.1: Quasi-Dynamic Domain State and Control Quadratized Device Model.....	13
Section 3.2: Quasi-Dynamic State and Control Algebraic Quadratic Companion Form	14
Section 3.3: Object-Oriented Modeling Example	17
Section 4: Automated Construction of Measurement Models.....	23
Section 4.1: Measurement Definitions	23
Section 4.2: Construction of the Measurement Model at Device Level	25
Section 4.3: Construction of the Measurement Model at Network Level.....	28
Section 4.4: Distribution System Distributed Quasi-Dynamic State Estimator (DS-DQSE)	29
4.4.1: DS-DQSE Algorithm.....	31
Section 5: Optimal Power Flow Formation.....	35
Section 5.1: Definition/Formation of Equality Constraints.....	36
Section 5.2: Construction of Inequality Constraints at Network Level	37
Section 5.3: Construction of Control Constraints at Network Level	39
Section 5.4: Construction of Objective Function at Network Level.....	39
Section 6: Optimal Power Flow Solution Algorithm	41
Section 6.1: Linearization	42
Section 6.2: Solution of the Linearized Problem.....	44
Section 6.3: Equality Equations Solution (Power Flow Problem)	46
Section 6.4: Iterative Linearization/Solution Method	47
Section 6.5: Determining Convergence or Addition of New Constraints	48
Section 7: Description of Example System	50
Section 8: Example DS-DQSE Results.....	55
Section 8.1: Measurement Creation.....	55
Section 8.2: QSE Implementation	58
Section 8.3: Test Case Results.....	62
Section 9: Summary and Conclusions.....	71
Appendix A: Object-Oriented Modeling Example (Converter with P-Q Control)	72

A.1: Converter Quasi-Dynamic Domain Compact Device Model.....	72
A.2: Converter Quasi-Dynamic Domain Quadratized Device Model	74
A.3: Converter Quasi-Dynamic Domain SCAQCF Device Model	76
Appendix B: Object-Oriented Algorithm of Constructing the Network SCAQCF Model from Device SCAQCF Models	79
Appendix C: Construction of Network SCAQCF Measurement Model	94
Appendix D: Device Operating Constraints	105
Appendix E: Construction of Equality Constraints in Quadratized OPF Problem	110
Appendix F: Construction of Inequality Constraints in Quadratized OPF Problem	113
Appendix G: Construction of Control Constraints in Quadratized OPF Problem.....	117
Appendix H: Construction of the Objective Function in Quadratized OPF Problem	118
Appendix I: Linearization of the Quadratized OPF Problem	122
H.1: Linearization of the Objective Function	122
H.2: Linearization of the Inequality Constraints	124

Section 1: Executive Summary

This report presents an object-oriented implementation of full state feedback control for virtual power plants (VPP). The components of the VPP full state feedback control are (1) object-oriented high-fidelity modeling for all devices in the VPP; (2) Distribution System Distributed Quasi-Dynamic State Estimation (DS-DQSE) that enables full observability of the VPP by augmenting actual measurements with virtual, derived and pseudo measurements and performing the Quasi-Dynamic State Estimation (QSE) in a distributed manner, and (3) automated formulation of the Optimal Power Flow (OPF) in real time using the output of the DS-DQSE, and solving the distributed OPF to provide the optimal control commands to the DERs of the VPP.

The infrastructure of this integrated system is the object-oriented high-fidelity device modeling within the monitoring devices of the VPP. The modeling approach starts from physically based models of power devices referred to as compact device models. Any existing model can be used as a compact device model. The compact model should be mathematically correct, meaning that the number of states and control variables should be consistent with the number of equations describing the compact model and the controls should be realizable. A quadratization procedure and the quadratic integration process are then applied to the compact device model, and the end result is an object-oriented, in a standardized syntax, interoperable model which is referred to as state and control algebraic quadratic companion form (SCAQCF). The DS-DQSE and OPF solvers work directly with the SCAQCF models without any other input (autonomous operation).

The second component of the approach is the DS-DQSE, a critical component for full state feedback control. The DS-DQSE provides in real-time the estimated states and validated models by performing QSE. The DS-DQSE is implemented in a distributed architecture where a distribution system (feeders) are partitioned into several sections. This partition is arbitrary with each section containing an arbitrary number of loads and resources, controllable or not. The DS-DQSE runs for each section. It requires that there is at least one local phasor measurement at each section. Given the measurements and the device SCAQCF models in a distribution system section, the DS-DQSE creates the measurement mathematical model at device-level. Then, with the help of network formation techniques, the measurement mathematical model from device-level are converted to network level measurement models. The state estimation algorithm works directly with the measurement mathematical models at the network level. The DS-DQSE provides a quantitative probabilistic consistency check between the network measurement model and the network model. Specifically, the DS-DQSE provides the best estimate of the states, the differences (residuals) between the measurements and the model predicted measurements as well as the expected standard deviation of these quantities. The DS-DQSE it also determines whether there are bad data and/or model discrepancies by the chi-square test. In case of such bad data, the source is identified by hypothesis testing. The overall process provides the best estimate of the state sand the validated model of the distribution section. Finally, the output of each DS-DQSE for each section is send to the distribution management system where the state and model of the

entire distribution system is constructed from the states of each section at a specific time stamp. We refer to it as the real time operating conditions and model.

The real time operating conditions and model (also in SCAQCF syntax) enables the optimal use of distributed energy resources (DER) units and provision of ancillary services incorporating operational constraints. This is achieved by automatically forming and solving an optimal power flow with appropriate objective. In this report, the objective is the levelization of the voltage profile along the distribution circuit. The formation of the OPF problem is automatic by simply using the objects of the network (in SCAQCF syntax) and the operating constraints (also in SCAQCF syntax). The automatically formulated OPF problem is then solved to provide the best settings of the various controls of the DERs as well as utility controls such as capacitor bank switching, tap changes, etc. The optimal power flow solution algorithm of the OPF solver is an iterative linear programming method. At each iteration, the OPF is linearized using the co-state method. The resulting linear optimization problem is in terms of only the control variables. The problem is converted to a linear program in standard form and solved to provide the optimal settings of the control variables. The process is repeated to convergence. A couple of iterations typically suffice. In the actual implementation, the computed optimal settings of the control variables can be transferred to the hardware that control the corresponding devices.

This report is organized as follows. Section 3 introduces the object-oriented high-fidelity device modeling approach. Section 4 illustrates the architecture and operation of DS-DQSE. Section 5 presents the definition and formation of the quadratized OPF problem. Section 6 presents the solution algorithm of the OPF problem. Section 7 presents an example test data for one section in the distribution system. Section 8 shows the example event data. Section 9 illustrates the implementation of DS-DQSE in a specific distribution system section. And section 10 summarizes the whole report.

Section 2: Introduction

The concept of the Virtual Power Plant (VPP) is quite general referring to collection of resources and power circuits that are under a coordinated control to make them behave as an entity which can respond to commands and behave as a controllable and dispatchable resource. A VPP can be a distribution system section with controllable loads and resources, a microgrid, etc. In this report, we focus on a distribution system section with resources and we focus on making this subsystem behave as a dispatchable plant by controlling the cluster of resources in this section. The report presents an object-oriented implementation of full state feedback control for VPPs. Figure 2.1 shows the integrated system of the VPP full state feedback control. An object-oriented method is used to represent models. Then, the Distribution System Distributed Quasi-Dynamic State Estimator (DS-DQSE) is applied to enable the extraction of the real time model and operating conditions of the VPP by performing Quasi-Dynamic State Estimation (QSE). Subsequently, an Optimal Power Flow is autonomously formulated and solved to provide the optimal controls. The optimal controls are send to the appropriate devices.

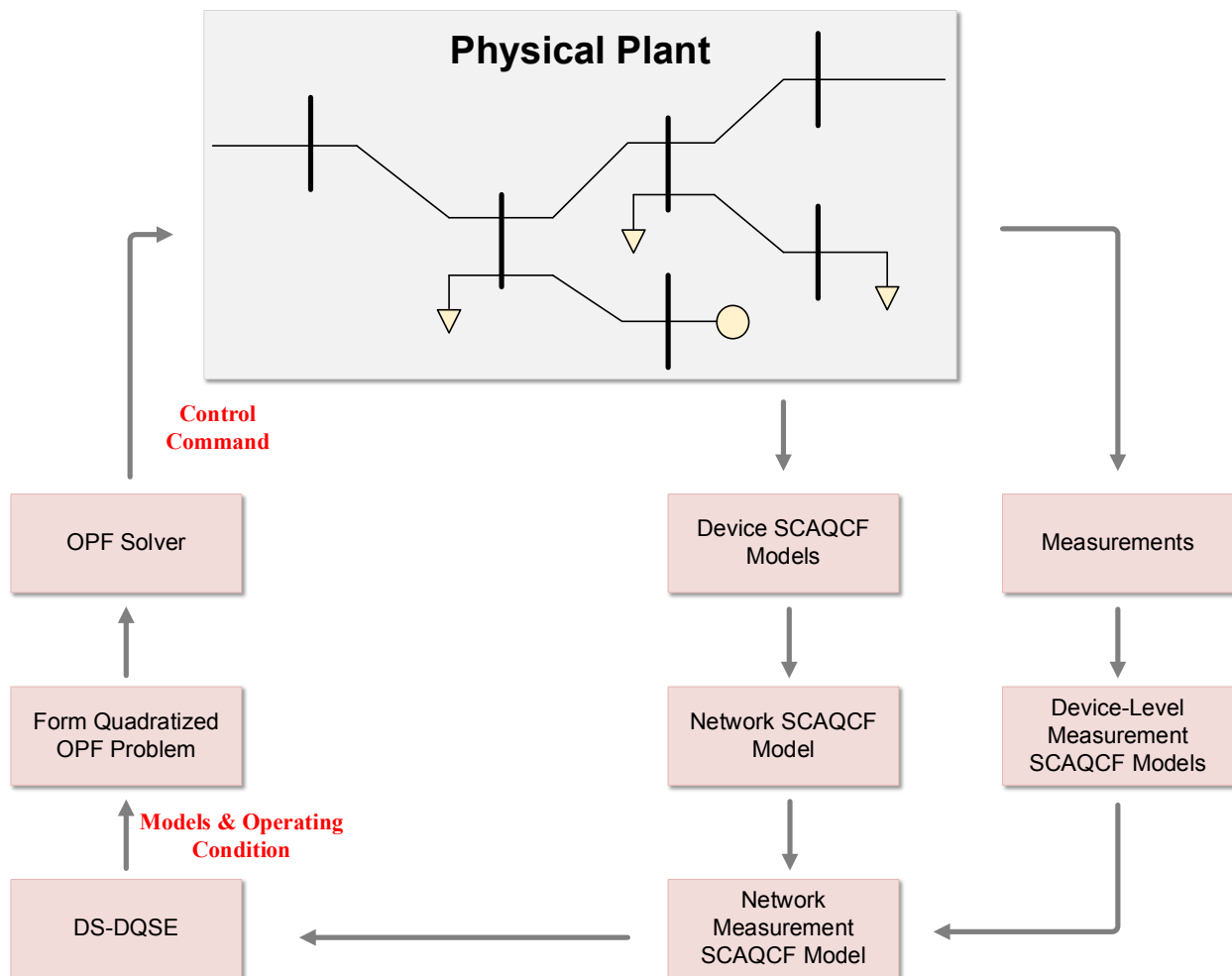


Figure 2.1: Integrated and Autonomous System of VPP Full State Feedback Control

The infrastructure for the integrated system is based on object-oriented high-fidelity device models for each device in the VPP. In this application, all device models are in quasi-dynamic domain, which ignore fast electromagnetic transients but include differential terms for slow dynamics such as those arising from electromechanical oscillations or the actions of a controller. The modeling approach starts from physically based models of power devices, referred to it as compact device models. Any existing model can be used as a compact device model, and these models are in terms of states and control variables. A quadratization procedure is then applied to the compact model if the compact model order is higher than two. This procedure consists of introducing additional variables to reduce higher order terms to nonlinear terms of highest order two. The result of this step is a quadratized device model in terms of state and control variables, which is referred as state and control quadratized device model (SCQDM). The SCQDM is then numerically integrated using the quadratic integration method for the purpose of converting it into an algebraic model that is referred to as the state and control algebraic quadratic companion form (SCAQCF). The syntax of the SCAQCF has been standardized and any power device can be converted into this form. The SCAQCF object is interoperable and usable by any application. For example, the DS-DQSE as well as the OPF formulator and solver work directly on the SCAQCF models without any other information.

The DS-DQSE requires measurements obtained on the system to perform the dynamic state estimation. Any measurement, irrespectively of the source of them measurements, i.e. actual, virtual, derived or pseudo, can be also expressed in the SCAQCF syntax. With increasing deployment of smart meters and other grid sensors in distribution systems, the amount of available measurements is growing. The measurements are expressed as functions of the state in the SCAQCF syntax and in this form are utilized by the DS-DQSE to perform a dynamic state estimation. The process of creating the measurement models in SCAQCF syntax is automated. Specifically, given the measurement set and all the SCAQCF device models, the measurement models are first developed at the device level, i.e. they are expressed as functions of the state variables of individual devices. Subsequently, the mapping between device states and system states is developed and the measurement models are converted from device level to system level. In this form the DS-DQSE performs a dynamic state estimation with the measurement models in terms of system state variables. The process is outlined in Figure 2.2. The dynamic state estimation includes an observability test, the actual state estimation and bad data detection and identification. Specifically, once the network SCAQCF measurement model is created, the DS-DQSE performs an observability test to determine that there are enough measurements to observe/compute the state. Subsequently it performs the dynamic state estimation and the chi-square test which checks the consistency between the estimated state and the network model. If this test indicates the presence of bad measurements, the DS-DQSE initiates the bad data identification and removes the bad data. The end result of the entire process is a validated model and a validated operating condition which can now be used for a variety of applications. In this report we will outline the application of optimizing the voltage profile of the feeder.

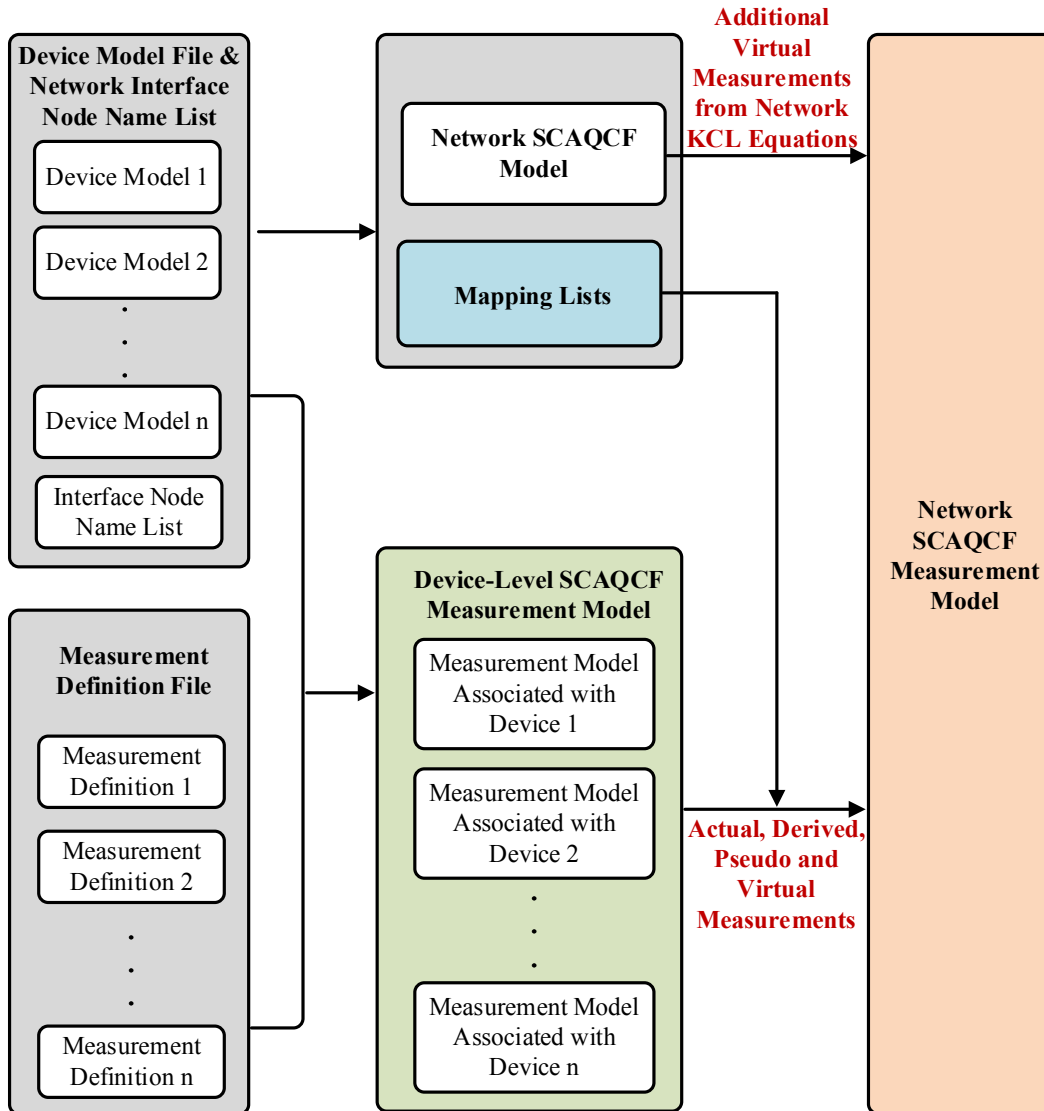


Figure 2.2: Flow Chart of Network SCAQCF Measurement Model Creation

As shown in Figure 2.3, the DS-DQSE is implemented in a distributed architecture. This is a novel approach compared to present available state estimation applications that are based on a centralized architecture and are executed in the control center. The distribution system (feeders) can be partitioned into several sections while each section containing some controllable loads and resources (i.e., each section is a VPP). The DS-DQSE is executed at each section of the feeder using local phasor measurements to perform DQSE for this local section. It is required that there should be at least one GPS synchronized measurement so that the computed best estimate of the state will have associated with it the time stamp for which this state estimate is valid. This is a critical requirement as the Distribution Energy Management System (DEMS) takes the state estimates for each distribution section with the exact time stamp and synthesizes the state estimate for the entire distribution system.

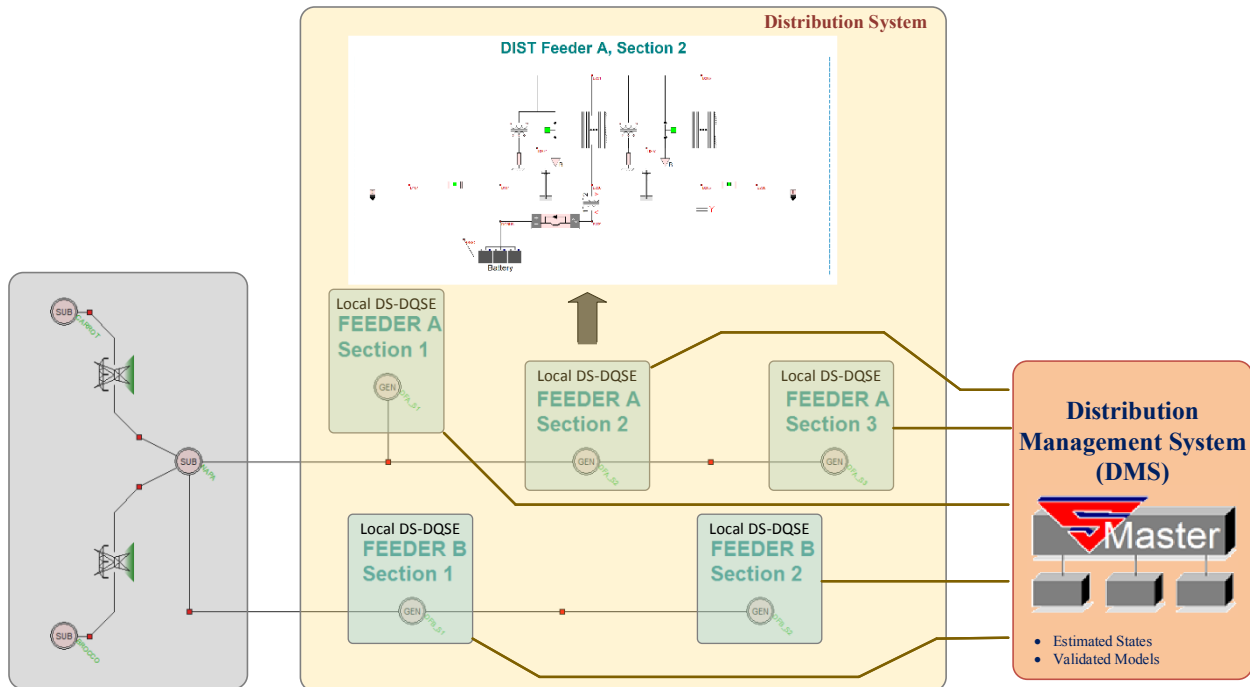


Figure 2.3: DS-DQSE for a Distribution System

The advantages of the distributed architecture are numerous. First of all, the state estimation algorithm is implemented using only local measurements to estimate the states in this local distribution section. Thus, the large data traffic is confined within the section, and the state estimator works on a small dimension subsystem compared to the one processed by a centralized state estimator. Secondly, since the dimension of the problem solved by DS-DQSE is significantly decreased, the execution time of the state estimator is fast (i.e. execution of once per cycle has been achieved). Thirdly, the relatively small dimension of the system allows very detailed power system models (three-phase dynamic models, instrumentation inclusive). The three-phase, instrumentation channel inclusive model for the power system can eliminate the estimation errors from the imbalanced operations and asymmetric models, as well as the measurement errors introduced by the instrumentation channels. In addition, because of the proposed measurement set, we increase the measurement redundancy of the distribution system section, which leads to more accurate estimation results. Last but not least, only the states and the validated model of each section are sent to the DEMS. This dramatically reduces the data communications and makes the whole state estimation system more efficient.

The DS-DQSE works as follows. Firstly, a data concentrator collects all the data from all IEDs in a specific section and converts and synchronizes these data into a C37.118 data stream. A local DS-DQSE is installed in this section and uses only the measurements from this section for the purpose of avoiding the requirement of obtaining and transmitting measurements via communication channels from other sections. Note that for this approach, data from at least one GPS-synchronized device is required in each section in order to synchronize all the data in the system. After the state estimation, the estimated states, and validated models for each section are

produced and sent to the Distribution Energy Management System (DEMS) where the system wide state estimate and model is synthesized.

The system wide state estimate and model, validated with the DQSE, is used to formulate and solve an Optimal Power Flow (OPF) to optimally control distributed energy resources (DER) units and/or provide ancillary services incorporating local network constraints. The objective function of the OPF can be user selected and the choices can be numerous. In this report, the objective is to improve the voltage profile along the distribution feeder. After defining the objective function, the formation of the OPF problem is automatic by simply using the object-oriented SCAQCF network model. As a matter of fact, the power flow equations of the model become the equality constraints of the OPF problem, and the operational constraints of the model become the inequality constraints of the OPF problem. Since all the equality and inequality constraints as well as the objective function is quadratic, the formulated OPF problem is a quadratized OPF problem. The general expression of the quadratized OPF problem is:

$$\begin{aligned}
\text{Minimize: } & J = Y_{objx}^T \mathbf{x} + Y_{obju}^T \mathbf{u} + \mathbf{x}^T F_{objxx} \mathbf{x} + \mathbf{u}^T F_{objuu} \mathbf{u} + \mathbf{u}^T F_{objux} \mathbf{x} + C_{objc} \\
\text{subject to: } & 0 = Y_{eqx} \mathbf{x} + Y_{equ} \mathbf{u} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T \langle F_{eqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{equ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{equx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} - B_{eq} - I \\
\text{where: } & B_{eq} = -N_{eqx} \mathbf{x}(t-h) - N_{equ} \mathbf{u}(t-h) - M_{eq} I(t-h) - K_{eq} \tag{2.1} \\
& Y_{ineqx} \mathbf{x} + Y_{inequ} \mathbf{u} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T \langle F_{ineqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{inequ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{inequx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + C_{ineqc} \leq 0 \\
& \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}
\end{aligned}$$

The automatically formulated quadratized OPF problem is solved by the OPF solver. The optimal power flow solution algorithm used in this report is briefly introduced as follows. The algorithm first uses the co-state method to linearize the OPF problem so that the OPF problem is converted into a linearized problem in terms of only control variables, i.e. the equality constraints (power flow) are used to eliminate the state variables. Subsequently, the linearized problem is converted into a linear program in standard form and it is solved with a simplex type algorithm. The computed control variables are inserted to the equality constraints which are solved to determine the new operating condition of the system. This is equivalent to a solution of the power flow problem. If the updated operating point violates any new constraints, then the violated constraint is added to the OPF problem and the process is repeated until convergence. The end result of the OPF solver is the optimal controls which are send to the appropriate devices.

The proposed OPF solution algorithm is robust and highly efficient. Robustness is achieved by virtue of starting from a feasible but not optimal solution and at each iteration the solution moves

the operating point in the feasible region while approaching the optimality. Therefore, at each iteration of the algorithm the solution iterate represents a feasible solution. High efficiency implies less runtime compared with traditional solution methods for the OPF problem. The reasons are as follows. Firstly, the algorithm models the OPF problem as a quadratic problem for fast convergence. Secondly, the algorithm identifies the active constraints gradually and adds them to the modeled constraint set if needed. These features of the algorithm ensure that at each iteration, the dimension of the problem is the smallest possible for the specific distribution system.

Section 3: Object-Oriented Device Modeling

This section describes a high-fidelity standardized modeling approach for power devices that enables object-oriented analysis in electric power systems.

As shown in Figure 3.1, the modeling approach starts from physical based models of power devices referred as compact device models. Any existing model can be used as a compact device model. In general, these models are in terms of states and control variables. A quadratization procedure is then applied to the compact model. This procedure consists of introducing additional variables to reduce higher order terms to nonlinear terms of highest order two. In case the compact model is linear or quadratic, this procedure is not needed. The end result is a quadratized device model which in general is also in terms of states and controls. The quadratized device model is integrated for the purpose of converting it into an algebraic model. We have selected the quadratic integration method for the integration. The reason for this selection is that the quadratic integration method has better properties than the popular trapezoidal integration method and it is also reasonably manageable (from the complexity point of view). The integration process transforms the state and control quadratized device model (SCQDM) into a state and control algebraic quadratic companion form (SCAQCF).

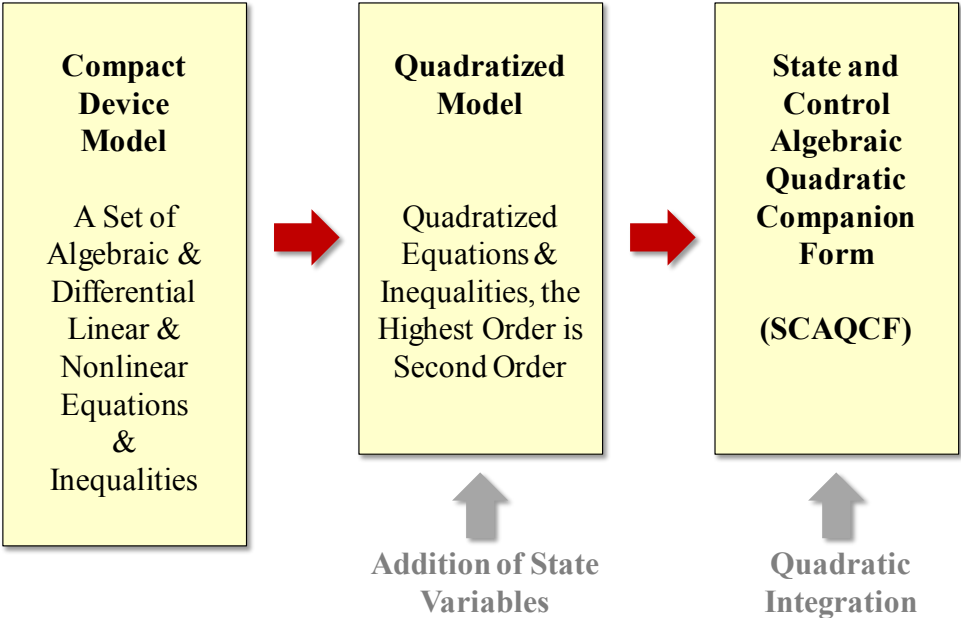


Figure 3.1: Object-Oriented Modeling Approach

It is also important to note that the models are in quasi-dynamic domain, where the compact models typically ignore fast electromagnetic transients but include differential terms for only slow dynamics such as those arising from electromechanical oscillations or controller actions.

This section is organized as follows: the quasi-dynamic domain SCQDM is described in Section 3.1, the quasi-dynamic domain SCAQCF device model is described in Section 3.2; and an example to illustrate the object-oriented modeling is described in Section 3.3.

Section 3.1: Quasi-Dynamic Domain State and Control Quadratized Device Model

The quasi-dynamic domain state and control quadratized device model (SCQDM) is used to represent the physical model and it is a preliminary step to obtain the quasi-dynamic State and Control Quadratic Companion Form (SCAQCF) device model. All the terms in SCQDM are at most second order. The specific syntax of the model is provided below with the following selections/requirements: (a) list all the linear equations for through variables first; (b) list all the remaining linear equations; (c) all differential terms only appear in the linear equations; (d) list all the remaining quadratic equations; (e) the equations containing through variables must be listed first; (f) the highest order of the model is second order. The requirements are always easily met by introduction of additional state variables. Note that the phasors are divided into real and imaginary parts in quadratized device model and that all the elements in the matrices are real values. The general expression for SCQDM is:

$$\begin{aligned}
I(t) &= Y_{eqx1} \mathbf{x}(t) + Y_{equ1} \mathbf{u}(t) + D_{eqxd1} \frac{d\mathbf{x}(t)}{dt} + C_{eqc1} \\
0 &= Y_{eqx2} \mathbf{x}(t) + Y_{equ2} \mathbf{u}(t) + D_{eqxd2} \frac{d\mathbf{x}(t)}{dt} + C_{eqc2} \\
0 &= Y_{eqx3} \mathbf{x}(t) + Y_{equ3} \mathbf{u}(t) + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}(t)^T \langle F_{eqxx3}^i \rangle \mathbf{x}(t) \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}(t)^T \langle F_{equu3}^i \rangle \mathbf{u}(t) \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}(t)^T \langle F_{equx3}^i \rangle \mathbf{x}(t) \\ \mathbf{M} \end{array} \right\} + C_{eqc3} \\
\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) &= Y_{fx} \mathbf{x}(t) + Y_{fu} \mathbf{u}(t) + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}(t)^T F_{fx}^i \mathbf{x}(t) \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}(t)^T F_{fu}^i \mathbf{u}(t) \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}(t)^T F_{fux}^i \mathbf{x}(t) \\ \mathbf{M} \end{array} \right\} + C_{fc}
\end{aligned}$$

Connectivity: *TerminalNodeName*

Normalization Factors: StateNormFactor, ThroughNormFactor, ControlNormFactor

$$\begin{aligned}
\text{subject to:} \quad & \mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{\max} \\
& \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}
\end{aligned}$$

where:

$I(t)$: the through variables of the device model;

$\mathbf{x}(t)$: external and internal state variables of the device model;

$\mathbf{u}(t)$: control variables of the device model, i.e. transformer tap, etc.;
 Y_{eqx1} : matrix defining the linear part for state variables in linear through variable equations;
 Y_{equ1} : matrix defining the linear part for control variables in linear through variable equations;
 D_{eqxd1} : matrices defining the differential part for state variables in linear through variable equations;
 C_{eqc1} : constant vector of the device model in linear through variable equations;
 Y_{eqx2} : matrix defining the linear part for state variables in linear virtual equations;
 Y_{equ2} : matrix defining the linear part for control variables in linear virtual equations;
 D_{eqxd2} : matrices defining the differential part for state variables in linear virtual equations;
 C_{eqc2} : constant vector of the device model in linear virtual equations;
 Y_{eqx3} : matrix defining the linear part for state variables in the remaining quadratic equations;
 Y_{equ3} : matrix defining the linear part for control variables in the remaining quadratic equations;
 C_{eqc3} : constant vector of the device model in the remaining quadratic equations;
 F_{eqxx} : matrices defining the quadratic part for state variables in the remaining quadratic equations;
 F_{equu} : matrices defining the quadratic part for control variables in the remaining quadratic equations;
 F_{eqxu} : matrices defining the quadratic part for the product of state and control variables in the remaining quadratic equations;
TerminalNodeName: terminal names defining the connectivity of the device model;
 StateNormFactor: Normalization Factors for the states;
 ThroughNormFactor: Normalization Factors for the through and zero variables;
 ControlNormFactor: Normalization Factors for the controls;
 $\mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{\max}$: operating constraints;
 $\mathbf{u}_{\min}, \mathbf{u}_{\max}$: lower and upper bounds for the control variables;

Y_{fx} : constraint matrix defining the linear part for state variables;
 F_{fx} : constraint matrices defining the quadratic part for state variables;
 Y_{fu} : constraint matrix defining the linear part for control variables;
 F_{fu} : constraint matrices defining the quadratic part for control variables;
 F_{fix} : constraint matrices defining the quadratic part for the product of state and control variables;
 C_f : constraint history dependent vector of the device model.

Section 3.2: Quasi-Dynamic State and Control Algebraic Quadratic Companion Form

The next step is to integrate the quasi-dynamic domain SCQDM model to derive an algebraic equivalent model. For this purpose the quadratic integration method is used. The end result is the quasi-dynamic domain State and Control Algebraic Quadratic Companion Form (SCAQCF). Note that this modeling standard can be applied to any device in the power system. The advantages of the SCAQCF device model are (a) it does not contain differential terms, it is algebraic, the dynamics are expressed in terms of past history terms, (b) the highest order is second order, and (c) it is easily cast into a standard syntax so that the utilization of the model can be performed by object oriented algorithms. The final expression for the quasi-dynamic domain SCAQCF device model is:

$$\begin{Bmatrix} \mathbf{I}(t) \\ 0 \\ 0 \\ \mathbf{I}(t_m) \\ 0 \\ 0 \end{Bmatrix} = Y_{eqx} \mathbf{x} + Y_{equ} \mathbf{u} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{M} \end{Bmatrix} \mathbf{x}^T \langle F_{eqx}^i \rangle \mathbf{x} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{M} \end{Bmatrix} \mathbf{u}^T \langle F_{equ}^i \rangle \mathbf{u} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{M} \end{Bmatrix} \mathbf{u}^T \langle F_{equx}^i \rangle \mathbf{x} - B_{eq}$$

$$B_{eq} = -N_{eqx} \mathbf{x}(t-h) - N_{equ} \mathbf{u}(t-h) - M_{eq} \mathbf{I}(t-h) - K_{eq}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) = Y_{feqx} \mathbf{x} + Y_{fequ} \mathbf{u} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{M} \end{Bmatrix} \mathbf{x}^T \langle F_{feqx}^i \rangle \mathbf{x} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{M} \end{Bmatrix} \mathbf{u}^T \langle F_{fequ}^i \rangle \mathbf{u} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{M} \end{Bmatrix} \mathbf{u}^T \langle F_{fequx}^i \rangle \mathbf{x} + C_{feq}$$

Connectivity: *TerminalNodeName*

$$\text{subject to: } \mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{\max}$$

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$$

Normalization Factor: StateNormFactor, ThroughNormFactor, ControlNormFactor

where

$I(t)$ and $I(t_m)$: the through variables of the device model;

\mathbf{x} : external and internal state variables of the device model, $\mathbf{x} = [\mathbf{x}(t), \mathbf{x}(t_m)]$;

\mathbf{u} : control variables of the device model, $\mathbf{u} = [\mathbf{u}(t), \mathbf{u}(t_m)]$;

Y_{eqx} : matrix defining the linear part for state variables;

F_{eqx}^i : matrices defining the quadratic part for state variables;

Y_{equ} : matrix defining the linear part for control variables;

F_{equ}^i : matrices defining the quadratic part for control variables;

F_{equx}^i : matrices defining the quadratic part for the product of state and control variables;

B_{eq} : history dependent vector of the device model;

N_{eqx} : matrix defining the last integration step state variables part;

N_{equ} : matrix defining the last integration step control variables part;

M_{eq} : matrix defining the last integration step through variables part;

K_{eq} : constant vector of the device model;

TerminalNodeName: terminal names defining the connectivity of the device model;

StateNormFactor: Normalization Factors for the states;

ThroughNormFactor: Normalization Factors for the through and zero variables;

ControlNormFactor: Normalization Factors for the controls;

$\mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{\max}$: operating constraints;

$\mathbf{u}_{\min}, \mathbf{u}_{\max}$: lower and upper bounds for the control variables;

Y_{feqx} : constraint matrix defining the linear part for state variables;

F_{feqx} : constraint matrices defining the quadratic part for state variables;

Y_{fequ} : constraint matrix defining the linear part for control variables;

F_{fequ} : constraint matrices defining the quadratic part for control variables;

F_{fequx} : constraint matrices defining the quadratic part for the product of state and control variables;

C_{feq} : constraint history dependent vector of the device model.

$$Y_{eqx} = \begin{bmatrix} \frac{4}{h} D_{eqxd1} + Y_{eqx1} & -\frac{8}{h} D_{eqxd1} \\ \frac{4}{h} D_{eqxd2} + Y_{eqx2} & -\frac{8}{h} D_{eqxd2} \\ Y_{eqx3} & 0 \\ \frac{1}{2h} D_{eqxd1} & \frac{2}{h} D_{eqxd1} + Y_{eqx1} \\ \frac{1}{2h} D_{eqxd2} & \frac{2}{h} D_{eqxd2} + Y_{eqx2} \\ 0 & Y_{eqx3} \end{bmatrix}$$

$$Y_{equ} = \begin{bmatrix} Y_{equ1} & 0 \\ Y_{equ2} & 0 \\ Y_{equ3} & 0 \\ 0 & Y_{equ1} \\ 0 & Y_{equ2} \\ 0 & Y_{equ3} \end{bmatrix}$$

$$F_{eqxx} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ F_{eqxx3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & F_{eqxx3} \end{bmatrix}$$

$$F_{equu} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ F_{equu3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & F_{equu3} \end{bmatrix}$$

$$F_{equx} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ F_{equx3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & F_{equx3} \end{bmatrix}$$

$$N_{eqx} = \begin{bmatrix} -Y_{eqx1} + \frac{4}{h} D_{eqxd1} \\ -Y_{eqx2} + \frac{4}{h} D_{eqxd2} \\ 0 \\ \frac{1}{2} Y_{eqx1} - \frac{5}{2h} D_{eqxd1} \\ \frac{1}{2} Y_{eqx2} - \frac{5}{2h} D_{eqxd2} \\ 0 \end{bmatrix} \quad N_{equ} = \begin{bmatrix} -Y_{equ1} \\ -Y_{equ2} \\ 0 \\ \frac{1}{2} Y_{equ1} \\ \frac{1}{2} Y_{equ2} \\ 0 \end{bmatrix} \quad M_{eq} = \begin{bmatrix} I_{size(i(t))} \\ 0 \\ 0 \\ -\frac{1}{2} I_{size(i(t))} \\ 0 \\ 0 \end{bmatrix} \quad K_{eq} = \begin{bmatrix} 0 \\ 0 \\ C_{eqc3} \\ \frac{3}{2} C_{eqc1} \\ \frac{3}{2} C_{eqc2} \\ C_{eqc3} \end{bmatrix}$$

Section 3.3: Object-Oriented Modeling Example

In this subsection, an IGBT-based converter average model with a P-Q controller is presented as an example of object-oriented device modeling. The compact model of the physical circuit, the quadratized model and the SCAQCF model are described respectively.

The diagram of the converter with a P-Q controller is demonstrated in Figure 3.2. The control variables of the system are the desired output active and reactive power (P_{ref} and Q_{ref}) of the converter. This can be achieved by controlling the modulation index of the converter and the phase angle difference between internal voltage \hat{E}_a^o and terminal voltage \hat{V}_a^o . The parameters of the converter model are the resistance on the DC side and the inductance on the AC side.

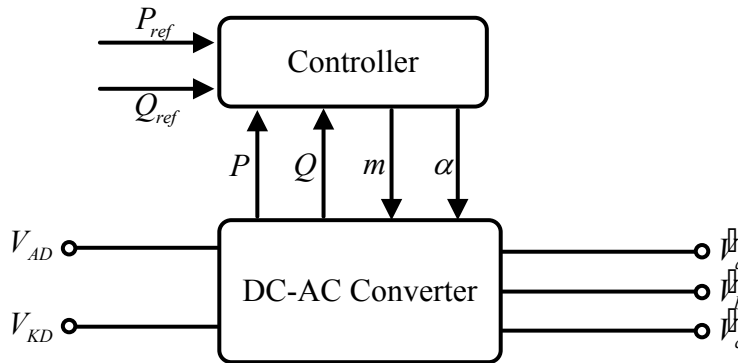


Figure 3.2: P-Q Control Converter

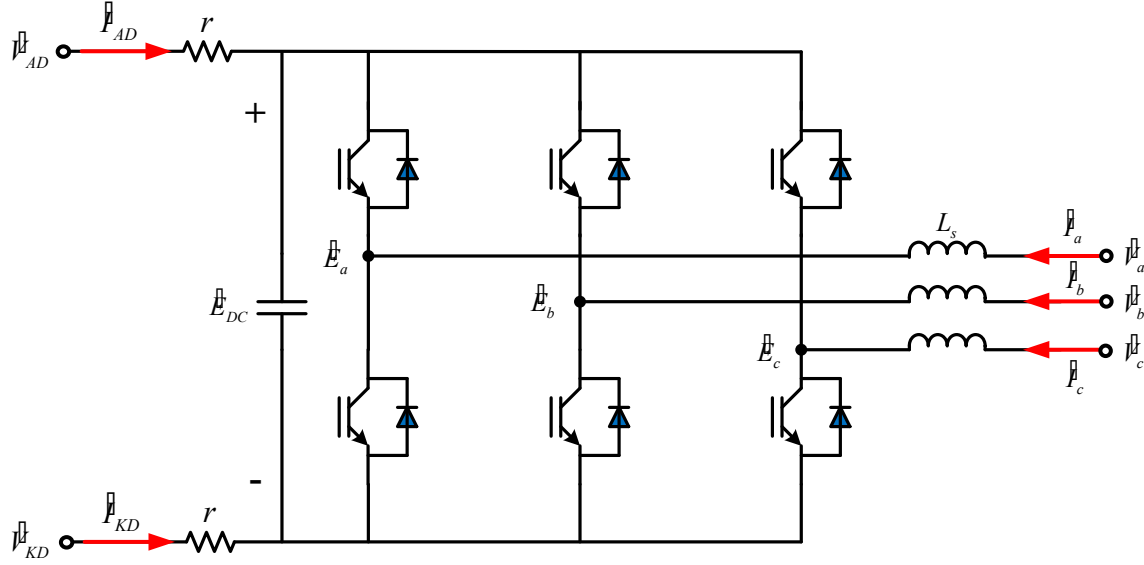


Figure 3.3: Circuit Diagram of the DC-AC Converter

A summary of this model in the standard form is as follows. First, the states are listed below.

State Index	Description of States	States	Units
0	Real part of V_{AD}°	V_{ADr}	kV
1	Imaginary part of V_{AD}°	V_{ADi}	kV
2	Real part of V_{KD}°	V_{KDr}	kV
3	Imaginary part of V_{KD}°	V_{KDi}	kV
4	Real part of V_a°	V_{ar}	kV
5	Imaginary part of V_a°	V_{ai}	kV
6	Real part of V_b°	V_{br}	kV
7	Imaginary part of V_b°	V_{bi}	kV
8	Real part of V_c°	V_{cr}	kV
9	Imaginary part of V_c°	V_{ci}	kV
10	Real part of E_{DC}°	E_{DCr}	kV
11	Imaginary part of E_{DC}°	E_{DCi}	kV
12	Real part of E_a°	E_{ar}	kV
13	Imaginary part of E_a°	E_{ai}	kV
14	Real part of E_b°	E_{br}	kV
15	Imaginary part of E_b°	E_{bi}	kV

16	Real part of \hat{E}_c	E_{cr}	kV
17	Imaginary part of \hat{E}_c	E_{ci}	kV
18	Real power output	P_{ac}	MW
19	Reactive power output	Q_{ac}	MVAr
20	Modulation index	m	No unit
21	Voltage magnitude of \hat{V}_a	V_{amag}	kV
22	Additional variable (modulation index times DC link voltage)	mE_{DC}	kV
23	Additional variable (mE_{DC} over V_{amag})	$mE_{DC}OverV$	No unit
24	Additional variable (sine function of the angle difference between \hat{E}_a and \hat{V}_a)	s_1	No unit
25	Additional variable (cosine function of the angle difference between \hat{E}_a and \hat{V}_a)	s_2	No unit

The control variables are:

Control Index	Description of controls	Controls	Units
0	Reference real power for P-Q controller	P_{ref}	MW
1	Reference reactive power for P-Q controller	Q_{ref}	MVAr

The parameters are:

Parameter Index	Description of Parameters	Parameter Variable	Default Setting
0	Converter equivalent resistance	r	0.03 ohm
1	Converter equivalent inductance	L	0.08 mH
2	Proportional coefficient of PQ controller for real power	PK_p	1.0

3	Integral coefficient of PQ controller for real power	PK_I	200.0
4	Proportional coefficient of PQ controller for reactive power	QK_p	1.0
5	Integral coefficient of PQ controller for reactive power	QK_I	200.0

The final equations for the model are listed below. The detailed derivation of this model is provided in Appendix A.

Equation Set 1 (linear through equations):

$$I_{ADr} = \frac{V_{ADr} - V_{KDr} - E_{DCr}}{2r} \quad (3.1)$$

$$I_{ADi} = \frac{V_{ADi} - V_{KDi} - E_{DCi}}{2r} \quad (3.2)$$

$$I_{KDr} = \frac{-V_{ADr} + V_{KDr} + E_{DCr}}{2r} \quad (3.3)$$

$$I_{KDi} = \frac{-V_{ADi} + V_{KDi} + E_{DCi}}{2r} \quad (3.4)$$

$$I_{ar} = \frac{1}{\omega L_s} (V_{ai} - E_{ai}) \quad (3.5)$$

$$I_{ai} = -\frac{1}{\omega L_s} (V_{ar} - E_{ar}) \quad (3.6)$$

$$I_{br} = \frac{1}{\omega L_s} (V_{bi} - E_{bi}) \quad (3.7)$$

$$I_{bi} = -\frac{1}{\omega L_s} (V_{br} - E_{br}) \quad (3.8)$$

$$I_{cr} = \frac{1}{\omega L_s} (V_{ci} - E_{ci}) \quad (3.9)$$

$$I_{ci} = -\frac{1}{\omega L_s} (V_{cr} - E_{cr}) \quad (3.10)$$

Equation Set 2 (linear internal equations):

$$0 = -\frac{1}{2}E_{ar} + \frac{\sqrt{3}}{2}E_{ai} - E_{br} \quad (3.11)$$

$$0 = -\frac{\sqrt{3}}{2}E_{ar} - \frac{1}{2}E_{ai} - E_{bi} \quad (3.12)$$

$$0 = -\frac{1}{2}E_{ar} - \frac{\sqrt{3}}{2}E_{ai} - E_{cr} \quad (3.13)$$

$$0 = \frac{\sqrt{3}}{2}E_{ar} - \frac{1}{2}E_{ai} - E_{ci} \quad (3.14)$$

$$0 = -K_{P1} \frac{dP_{ac}}{dt} + K_{I1} (P_{ref} - P_{ac}) - \frac{ds_1}{dt} \quad (3.15)$$

$$0 = -K_{P2} \frac{dQ_{ac}}{dt} + K_{I2} (Q_{ref} - Q_{ac}) - \frac{dm}{dt} \quad (3.16)$$

Equation Set 3 (quadratic equations):

$$0 = \frac{1}{2r} (V_{ADr} E_{DCr} - V_{KDr} E_{DCr} - E_{DCr}^2) - P_{ac} \quad (3.17)$$

$$0 = E_{DCi} \quad (3.18)$$

$$0 = \frac{1}{\omega L_s} (-V_{ar} E_{ai} + V_{ai} E_{ar} - V_{br} E_{bi} + V_{bi} E_{br} - V_{cr} E_{ci} + V_{ci} E_{cr}) + P_{ac} \quad (3.19)$$

$$0 = \frac{1}{\omega L_s} (V_{ar}^2 - V_{ar} E_{ar} + V_{ai}^2 - V_{ai} E_{ai} + V_{br}^2 - V_{br} E_{br} + V_{bi}^2 - V_{bi} E_{bi} + V_{cr}^2 - V_{cr} E_{cr} + V_{ci}^2 - V_{ci} E_{ci}) + Q_{ac} \quad (3.20)$$

$$0 = m \cdot E_{DC} - m E_{DC} \quad (3.21)$$

$$0 = V_{ar}^2 + V_{ai}^2 - V_{a,mag}^2 \quad (3.22)$$

$$0 = m E_{DC} - V_{a,mag} \cdot m E_{DC} \text{ Over } V \quad (3.23)$$

$$0 = \frac{1}{2\sqrt{2}} m E_{DC} \text{ Over } V \cdot V_{ar} - E_{ar} \cdot s_2 - E_{ai} \cdot s_1 \quad (3.24)$$

$$0 = \frac{1}{2\sqrt{2}} m E_{DC} \text{ Over } V \cdot V_{ai} - E_{ai} \cdot s_2 + E_{ar} \cdot s_1 \quad (3.25)$$

$$0 = s_1^2 + s_2^2 - 1.0 \quad (3.26)$$

Operation Constraints:

$$-I_{DC,\max} \leq \frac{1}{2r}V_{ADr} - \frac{1}{2r}V_{KDr} - \frac{1}{2r}E_{DCr} \leq I_{DC,\max} \quad (3.c1)$$

$$0 \leq \frac{1}{\omega^2 L_s^2}V_{ar}^2 + \frac{1}{\omega^2 L_s^2}V_{ai}^2 + \frac{1}{\omega^2 L_s^2}E_{ar}^2 + \frac{1}{\omega^2 L_s^2}E_{ai}^2 - \frac{2}{\omega^2 L_s^2}V_{ar}E_{ar} - \frac{2}{\omega^2 L_s^2}V_{ai}E_{ai} \leq I_{AC,\max}^2 \quad (3.c2)$$

$$0 \leq \frac{1}{\omega^2 L_s^2}V_{br}^2 + \frac{1}{\omega^2 L_s^2}V_{bi}^2 + \frac{1}{\omega^2 L_s^2}E_{br}^2 + \frac{1}{\omega^2 L_s^2}E_{bi}^2 - \frac{2}{\omega^2 L_s^2}V_{br}E_{br} - \frac{2}{\omega^2 L_s^2}V_{bi}E_{bi} \leq I_{AC,\max}^2 \quad (3.c3)$$

$$0 \leq \frac{1}{\omega^2 L_s^2}V_{cr}^2 + \frac{1}{\omega^2 L_s^2}V_{ci}^2 + \frac{1}{\omega^2 L_s^2}E_{cr}^2 + \frac{1}{\omega^2 L_s^2}E_{ci}^2 - \frac{2}{\omega^2 L_s^2}V_{cr}E_{cr} - \frac{2}{\omega^2 L_s^2}V_{ci}E_{ci} \leq I_{AC,\max}^2 \quad (3.c4)$$

$$0.0 \leq m \leq 1.0 \quad (3.c5)$$

Section 4: Automated Construction of Measurement Models

This section introduces the computational procedure which enables data from sensors to be steamed and used by the distributed quasi-dynamic state estimation. With increasing deployment of smart meters and other grid sensors in distribution systems, the amount of available measurements is growing. These measurements as well as the other measurements proposed in Section 4.1 form the DS-DQSE measurement set that enables the estimation of the distribution system operating state. Given the measurement set and all the device models in a distribution system section, the DS-DQSE creates the measurement models at device-level, i.e. the measurements are expressed as a function of the device states. Then, a network formation algorithm creates the mapping between the states of individual devices to the state of the network. Using the mapping, each measurement model is transformed into a model in terms of the network states. In this form, the measurements are used to perform a dynamic state estimation and provide the best estimate of the network states. The dynamic state estimation basically quantifies the consistency between the measurements and the network model. The estimated states and the validated model for the whole distribution system section together with a quantitative confidence level for the validity of the model and states is provided to the distribution management system. This output information from the DS-DQSE can be used for any application that requires the real time model and operating conditions of the VPP.

The organization of this section is as follows. Section 4.1 describes the measurement definition set for DS-DQSE. Section 4.2 introduces an object-oriented way to create the device-level measurement model. Section 4.3 describes the network-level measurement model creation. And Section 4.4 illustrates the algorithm of distributed quasi-dynamic state estimation.

Section 4.1: Measurement Definitions

With increasing deployment of smart meters and other grid sensors in distribution systems, the amount of available measurements is growing. These measurements enable implementation of distribution system state estimators to provide real-time models and operating conditions of the distribution network. To further increase redundancy and accuracy of the estimated states, we propose the state estimator measurement definition set where the measurements are classified into four types:

- (a) actual measurements: measurements from actual measurement channels, i.e., any measurements from any IEDs (relays, meters, FDR, PMUs, etc.);
- (b) derived measurements: measurements derived from actual measurements based on topology. Figure 4.1 shows an example of creating a derived measurement in a distribution system section. In the figure, three-phase current measurements from B13 to B14 and three-phase current

measurements of the capacitor bank at B25 are available. Thus, as shown in equation (4.1), the three-phase current from B13 to B12 is computed by applying Kirchhoff's current law (KCL), which is treated as a derived measurement.

$$I_{B13_B12,abc}^{\%} = - \left(I_{B12_B13,abc}^{\%} + I_{B25,abc}^{\%} \right) \quad (4.1)$$

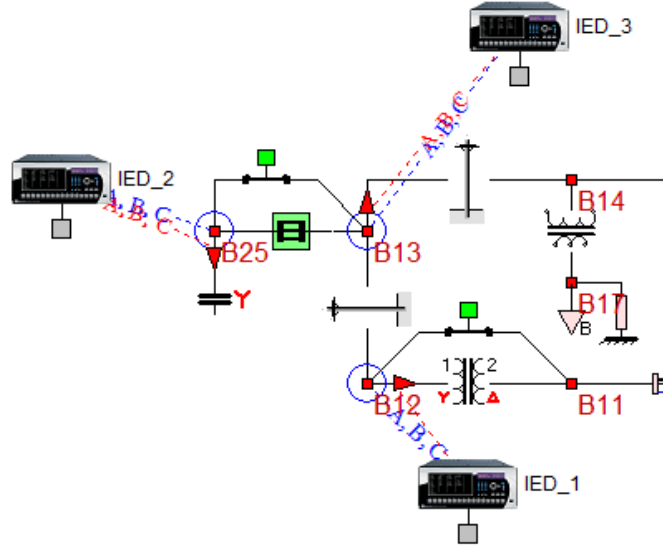


Figure 4.1: Example of a Derived Measurement

(c) virtual measurements: mathematical quantities defined by physical laws, such as KCL, model internal equations, etc. Figure 4.2 shows an example of creating a virtual measurement in a distribution system section. In the figure, three-phase current measurements from B301 to B300 and three-phase current measurements from B301 to B302 are available. According to KCL, the sum of these two three-phase current measurements at B301 is zero, which is treated as a virtual measurement as shown in equation (4.2).

$$0 = I_{B301_B300,abc}^{\%} + I_{B301_B302,abc}^{\%} \quad (4.2)$$

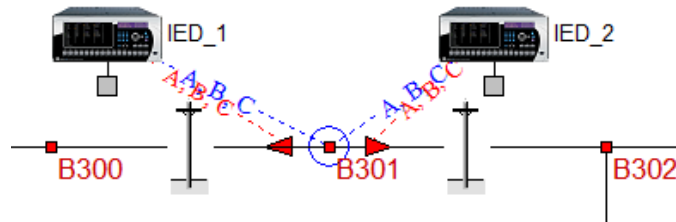


Figure 4.2: Example of a Virtual Measurement

(d) pseudo measurements: not directly measured, represent quantities for which their values are approximately known, such as missing phase measurements, neutral/shield voltage measurements, neutral currents, etc.

Section 4.2: Construction of the Measurement Model at Device Level

The construction of the network measurement model consists of two steps. The first step is to use the given device model file and the measurement definition file to create the SCAQCF measurement models associated with each device. These device-level measurement models contain device-level actual, derived, pseudo and virtual measurements. The second step is to construct the network SCAQCF measurement model from device-level measurement models. This step is achieved by first using the given device model file and the network interface node name list to create the network SCAQCF model and the mapping lists. Then we create the network measurement model from device-level measurement model via mapping lists while adding additional virtual measurements (network KCL equations) from the network SCAQCF model. The whole procedure is shown in Figure 2.2.

This subsection introduces the procedure to create the device-level SCAQCF measurement models from measurement definitions as described in Section 4.1. The problem is stated as follows. Given all the devices in the network and all the measurement definitions from each device, construct the device measurement model in SCAQCF syntax. The construction must be performed automatically. The construction of the device measurement model is illustrated below.

Actual Across Measurement:

An actual across measurement of one device is a linear combination of state variables of this device, i.e.

$$z(t) = A\mathbf{x}(t) + \eta,$$

where $z(t)$ is the measurement, A is the linear coefficient matrix, $\mathbf{x}(t)$ is the device state vector, and η is the noise error provided by the meter.

Actual Through Measurement:

The actual through measurement equation is obtained from the device model. For instance, if there is a current measurement at the j th terminal of a device, then the measurement model is the equation corresponding to the j th terminal in this device model, i.e.

$$z(t) = Y_{zx}\mathbf{x}(t) + Y_{zu}\mathbf{u}(t) + \begin{Bmatrix} \mathbf{M} \\ \mathbf{x}^T F_{zx}^i \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T F_{zu}^i \mathbf{u} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T F_{zux}^i \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + N_{zx}\mathbf{x}(t-h) + N_{zu}\mathbf{u}(t-h) + M_z i(t-h) + K_z + \eta$$

where $z(t)$ is the measurement, Y_{zx} is the linear coefficient matrix for state variables, Y_{zu} is the linear coefficient matrix for control variables, F_{zx}^i is the quadratic part for state variables, F_{zu}^i is

the quadratic part for control variables , F_{zux}^i is the quadratic part for the product of state and control variables , N_{zx} is the linear coefficient matrix for past history state variables, N_{zu} is the linear coefficient matrix for past history control variables, M_z is the linear coefficient matrix for past history through variables, K_z is the constant value, and η is the noise error provided by the meter.

Derived States Measurements:

A derived state measurement of one device is a linear combination of state variables of this device, i.e.

$$z(t) = A\mathbf{x}(t) + \eta ,$$

where $z(t)$ is the measurement, A is the linear coefficient matrix, $\mathbf{x}(t)$ is the device state vector, and η is the noise error provided by the meter.

Derived Functional Measurements:

The derived functional measurement equation is obtained from the device model. For instance, if there is a derived current measurement at the j th terminal of a device, then the measurement model is the equation corresponding to the j th terminal in this device model, i.e.

$$z(t) = Y_{zx}\mathbf{x}(t) + Y_{zu}\mathbf{u}(t) + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T F_{zx}^i \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T F_{zu}^i \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T F_{zux}^i \mathbf{x} \\ \mathbf{M} \end{array} \right\} + N_{zx}\mathbf{x}(t-h) + N_{zu}\mathbf{u}(t-h) + M_z i(t-h) + K_z + \eta$$

where $z(t)$ is the measurement, Y_{zx} is the linear coefficient matrix for state variables, Y_{zu} is the linear coefficient matrix for control variables, F_{zx}^i is the quadratic part for state variables , F_{zu}^i is the quadratic part for control variables , F_{zux}^i is the quadratic part for the product of state and control variables , N_{zx} is the linear coefficient matrix for past history state variables, N_{zu} is the linear coefficient matrix for past history control variables, M_z is the linear coefficient matrix for past history through variables, K_z is the constant value, and η is the noise error provided by the meter.

Virtual Measurements:

Virtual Measurements are those that express physical or mathematical laws such as Kirchhoff Current Law. For instance, the zero sum of the currents at a common node is a virtual measurement.

$$0 = Y_{zx} \mathbf{x}(t) + Y_{zu} \mathbf{u}(t) + \begin{Bmatrix} \mathbf{M} \\ \mathbf{x}^T F_{zx}^i \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T F_{zu}^i \mathbf{u} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T F_{zux}^i \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + N_{zx} \mathbf{x}(t-h) + N_{zu} \mathbf{u}(t-h) + M_z i(t-h) + K_z + \eta$$

where Y_{zx} is the linear coefficient matrix for state variables, Y_{zu} is the linear coefficient matrix for control variables, F_{zx}^i is the quadratic part for state variables, F_{zu}^i is the quadratic part for control variables, F_{zux}^i is the quadratic part for the product of state and control variables, N_{zx} is the linear coefficient matrix for past history state variables, N_{zu} is the linear coefficient matrix for past history control variables, M_z is the linear coefficient matrix for past history through variables, K_z is the constant value, and η is the noise error.

Pseudo State Measurements:

A pseudo state measurement of one device is a linear combination of state variables of this device, i.e.

$$z(t) = A \mathbf{x}(t) + \eta$$

where $z(t)$ is the measurement, A is the linear coefficient matrix, $\mathbf{x}(t)$ is the device state vector, and η is the noise error of this pseudo measurement.

Pseudo Functional Measurements:

The pseudo functional measurement equation is obtained from the device model. For instance, if there is a pseudo current measurement at the j th terminal of a device, then the measurement model is the equation corresponding to the j th terminal in this device model, i.e.

$$z(t) = Y_{zx} \mathbf{x}(t) + Y_{zu} \mathbf{u}(t) + \begin{Bmatrix} \mathbf{M} \\ \mathbf{x}^T F_{zx}^i \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T F_{zu}^i \mathbf{u} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T F_{zux}^i \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + N_{zx} \mathbf{x}(t-h) + N_{zu} \mathbf{u}(t-h) + M_z i(t-h) + K_z + \eta$$

where $z(t)$ is the measurement, Y_{zx} is the linear coefficient matrix for state variables, Y_{zu} is the linear coefficient matrix for control variables, F_{zx}^i is the quadratic part for state variables, F_{zu}^i is the quadratic part for control variables, F_{zux}^i is the quadratic part for the product of state and control variables, N_{zx} is the linear coefficient matrix for past history state variables, N_{zu} is the linear coefficient matrix for past history control variables, M_z is the linear coefficient matrix for past history through variables, K_z is the constant value, and η is the noise error of this pseudo measurement.

The measurement models at the device-level can be expressed as a vector function with the following general expression. Note that the general expression below becomes a part of the device object (the SCAQCF object).

$$\mathbf{z} = Y_{devm,x} \mathbf{x}(t) + Y_{devm,u} \mathbf{u}(t) + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T F_{devm,x}^i \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T F_{devm,u}^i \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T F_{devm,xu}^i \mathbf{x} \\ \mathbf{M} \end{array} \right\} + C_{devm} + \boldsymbol{\eta}$$

$$C_{devm} = N_{devm,x} \mathbf{x}(t-h) + N_{devm,u} \mathbf{u}(t-h) + M_{devm} \mathbf{i}(t-h) + K_{devm}$$

Measurement noise error: dMeterScale, dMeterSigmaPU

where:

\mathbf{z} : measurement variables at both time t and time t_m , $\mathbf{z} = [\mathbf{z}(t), \mathbf{z}(t_m)]$;

\mathbf{x} : external and internal state variables of the measurement model, $\mathbf{x} = [\mathbf{x}(t), \mathbf{x}(t_m)]$;

\mathbf{u} : control variables of the measurement model, i.e. transformer tap, etc. $\mathbf{u} = [\mathbf{u}(t), \mathbf{u}(t_m)]$;

$Y_{devm,x}$: matrix defining the linear part for state variables of the device-level measurement model;

$F_{devm,x}$: matrices defining the quadratic part for state variables of the device-level measurement model;

$Y_{devm,u}$: matrix defining the linear part for control variables of the device-level measurement model;

$F_{devm,u}$: matrices defining the quadratic part for control variables of the device-level measurement model;

$F_{devm,xu}$: matrices defining the quadratic part for the product of state and control variables of the device-level measurement model;

C_{devm} : history dependent vector of the device-level measurement model;

$N_{devm,x}$: matrix defining the last integration step state variables part of the device-level measurement model;

$N_{devm,u}$: matrix defining the last integration step control variables part of the device-level measurement model;

M_{devm} : matrix defining the last integration step through variables part of the device-level measurement model;

K_{devm} : constant vector of the measurement model of the device-level measurement model;

$dMeterScale$: the scale that meters use (in metric units);

$dMeterSigmaPU$: the standard deviation for the measurements (in per. unit).

Section 4.3: Construction of the Measurement Model at Network Level

This section introduces the procedure to create the network-level SCAQCF measurement model. This task is achieved by two subtasks: (1) Create the network model of this distribution system section and the mapping lists from devices to this network; (2) Use the mapping lists to create the network-level SCAQCF measurement model from device-level SCAQCF measurement models and add the network KCL equations as additional virtual measurements to the network

measurement model. Figure 2.2 shows the flow chart of network-level SCAQCF measurement model construction. And the general procedure is described here.

The first task is to form the network SCAQCF model. The purpose of the network formation is to (1) provide the mapping lists (states, equations, controls, and constraints) from devices to the network, and (2) provide the network KCL equations at the common nodes. Notice that the formation procedure is object-oriented, in other words, given all the device SCAQCF models in this network and the network interface node name list, the results are the automatically constructed network SCAQCF model and the mapping lists. Appendix B illustrates the detailed object-oriented algorithm for constructing the network SCAQCF model and its SCAQCF expression.

The next step is to form the network SCAQCF measurement model. This task is achieved by using the mapping information to transform the measurement model from device-level to network-level. Specifically, given the network SCAQCF model and the mapping lists, the network SCAQCF measurement model is automatically constructed. It is accomplished by the following two subtasks: (1) Use the mapping lists (device states to network states, device equations to network equations, and device controls to network controls), the states and controls in the device-level measurement models are replaced with network-level states and controls; (2) Add network KCL equations as additional virtual measurements to the network-level SCAQCF measurement model. The detailed object-oriented algorithm for construction of network SCAQCF measurement model appears in Appendix C.

Section 4.4: Distribution System Distributed Quasi-Dynamic State Estimator (DS-DQSE)

This section introduces the architecture and the algorithm of DS-DQSE. As a distribution system section with a cluster of controllable loads and resources, VPP acts as a critical role in the distribution system operation and control. To optimal control the VPP, the accurate operating condition and accurate distribution system model are required. And DS-DQSE is able to solve this problem.

DS-DQSE has following characteristics to fit and support the VPP: (a) State estimation and data validation: DS-DQSE provides real-time estimated states, validated measurements, and validated models through distributed dynamic state estimation. Notice that in addition to the actual data collected from IEDs, several other types of measurements are defined, resulting in high measurement redundancy. Such high redundancy guarantees the accuracy of the estimated states and the network model of VPP. (b) Anomalies detection and root cause identification: the hidden failures such as blown fuses, cut wires, etc. or human errors such as incorrect entry of system parameters such as CT and VT ratios, incorrect instrument transformer connection (delta/wye) can be detected and identified. (c) Missing data creation: the missing data can be estimated and created in case of temporary loss of data.

As shown in Figure 4.4, the DS-DQSE is implemented in a distributed architecture. This is a novel approach compared to present available state estimation applications that are based on a centralized architecture and are executed in the control center. The distribution system (feeders) can be partitioned into several sections while each section containing some controllable loads and resources (i.e., each section is a VPP). Each section installs a DS-DQSE to perform Quasi-Dynamic State Estimation (QSE) for this local section. QSE incorporates slow dynamics (e.g., electromechanical transients of rotating electrical machines, controls of power electronics, etc.) while neglecting fast electromagnetic transients. The advantage of the distributed architecture is numerous. First of all, the DS-DQSE is implemented using only local measurements to estimate the states of the local distribution section. Thus, the data traffic is confined, and the state estimator works on a small dimension of the system compared to the one processed by a centralized state estimator. Secondly, since the dimensionality of the problem solved by DS-DQSE is significantly decreased, the execution time of the state estimator is fast (i.e. at each cycle). Thirdly, the relative small dimension of the system allows very detailed power system models (three-phase dynamic models, instrumentation inclusive). The three-phase, instrumentation channel inclusive model can eliminate the estimation errors from the imbalanced operations and asymmetric system, as well as the measurement errors introduced by the instrumentation channels. In addition, because of the proposed measurement set, we increase the measurement redundancy of the distribution system section and therefore more accurate estimation results. Last but not least, only the states and validated models of each section are sent to the distribution energy management system (DEMS). This feature dramatically reduces the data communications and makes the whole state estimation system more efficient.

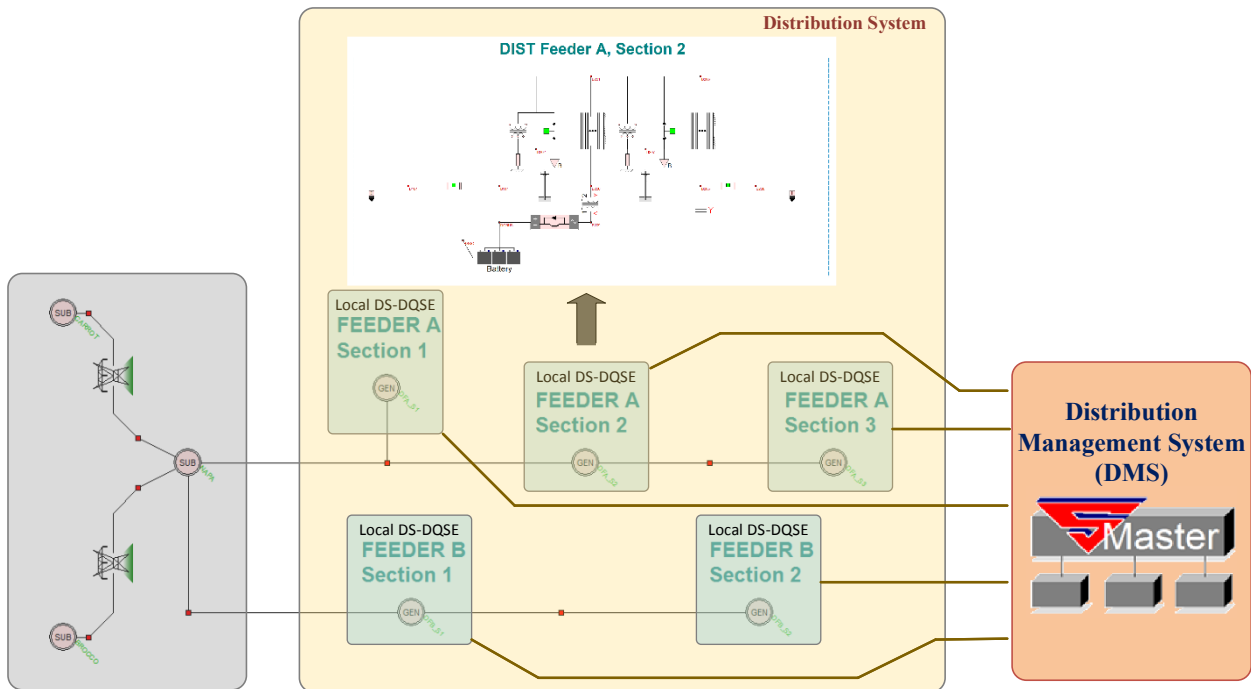


Figure 4.4: DS-DQSE for a Distribution System

The DS-DQSE works as follows. Firstly, a data concentrator collects all the data from all IEDs in a specific section and converts and synchronizes these data into a C37.118 data stream. A local DS-DQSE is installed in this section and only uses all the measurements from this section for the purpose of avoiding the requirement of obtaining and transmitting measurements via communication channels from other sections. Note that for this approach, data from at least one GPS-synchronized device is required in each section in order to time tag the estimation results with GPS accuracy. After the state estimation, the estimated states, validated measurements, and validated models for each section are sent to the DMS where the system wide estimated states and model are synthesized.

The estimator is defined in terms of models, states, measurement sets and estimation methods. The quasi-dynamic state estimation algorithm is object-oriented, i.e. all the models in the system are expressed in SCAQCF syntax (described in Section 3) and the DS-DQSE operates directly on these object models. The local state estimator uses the generated network-level SCAQCF measurement models (see previous section) to perform QSE and outputs the estimated states, validated measurements, and validated models. This approach allows efficient bad data detection and identification, alarm analysis and root cause identification. The advantage comes from the fact that in each local section, the DS-DQSE has greater redundancy of data compared to a typical centralized state estimator based on SCADA data alone.

4.4.1: DS-DQSE Algorithm

The DS-DQSE uses three different methods to estimate the states: (a) Unconstrained Least Square Method, (b) Constrained Least Square Method, and (c) Extended Kalman Filtering Method. The unconstrained weighted least square (UWLS) method is briefly presented below.

From section 4.3, we have the network measurement model:

$$\mathbf{z} = Y_{netm,x} \mathbf{x}(t) + Y_{netm,u} \mathbf{u}(t) + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T F_{netm,x}^i \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T F_{netm,u}^i \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T F_{netm,ux}^i \mathbf{x} \\ \mathbf{M} \end{array} \right\} + C_{netm} + \boldsymbol{\eta} \quad (4.3)$$

$$C_{netm} = N_{netm,x} \mathbf{x}(t-h) + N_{netm,u} \mathbf{u}(t-h) + M_{netm} \mathbf{i}(t-h) + K_{netm}$$

For a given state estimation, it is assumed that the controls do not change during this short period and therefore are treated as constants. Therefore the measurements \mathbf{z} are expressed as functions of the states:

$$\mathbf{z} = Y_{netm,x} \mathbf{x} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T F_{netm,x}^i \mathbf{x} \\ \mathbf{M} \end{array} \right\} - C_{netm} + \boldsymbol{\eta} = h(\mathbf{x}) + \boldsymbol{\eta} \quad (4.4)$$

$$C_{netm} = N_{netm,x} \mathbf{x}(t-h) + M_{netm} \mathbf{i}(t-h) + K_{netm}$$

where \mathbf{z} is the measurement vector of the system, $Y_{netm,x}$ is the linear coefficient matrix regarding to the state vector \mathbf{x} , $F_{netm,x}^i$ is the nonlinear (quadratic) coefficient matrix, C_{netm} is the history dependent vector, $N_{netm,x}$ is the linear coefficient matrix regarding to the last integration step state variables, M_{netm} is the linear coefficient matrix regarding to the last integration step through, K_{netm} is the constant vector of the network measurement model, and $\boldsymbol{\eta}$ is the measurement error.

The standard deviation (the measurement error) of each measurement is part of the measurement data and depend on the IED from which the data have been obtained. The pseudo-measurements are not associated with any physical IED and their standard deviations are set as a relatively high value (e.g., 0.1 p.u.). Virtual measurements are measurements with zero standard deviation. To avoid numerical problems, a relatively small standard deviation is used (e.g., 0.001 p.u.).

The UWLS method minimizes the sum of the weighted squares of the components of the residual vector. Mathematically:

$$\text{Minimize } J = (\mathbf{z}(\mathbf{t}) - h(\mathbf{x}))^T \mathbf{W} (\mathbf{z}(\mathbf{t}) - h(\mathbf{x})) \quad (4.5)$$

where \mathbf{W} is the weight matrix with the weights defined as the inverse of the squared standard deviations: $\mathbf{W} = \text{diag} \{1/\sigma_1^2, 1/\sigma_2^2, \dots, 1/\sigma_n^2\}$, and σ_i is the standard deviation corresponding to each measurement z_i .

Unknown state vector \mathbf{x} is obtained by the optimal condition:

$$dJ/d\mathbf{x} = 0 \quad (4.6)$$

To obtain the solution of the nonlinear optimization problem above, we linearize the nonlinear equations (the highest order is the second order in the measurement model) at the point \mathbf{x}^v by assuming that an initial guess \mathbf{x}^v is very close to the optimal solution:

$$\mathbf{r} = h(\mathbf{x}^v) + \partial h(\mathbf{x}) / \partial \mathbf{x} \Big|_{\mathbf{x}=\mathbf{x}^v} (\mathbf{x} - \mathbf{x}^v) - \mathbf{z} \quad (4.7)$$

After we set

$$\mathbf{H} = \partial h(\mathbf{x}) / \partial \mathbf{x} \Big|_{\mathbf{x}=\mathbf{x}^v}, \quad (4.8)$$

and

$$\mathbf{z}' = -h(\mathbf{x}^v) + \mathbf{H}\mathbf{x}^v + \mathbf{z}, \quad (4.9)$$

the equation becomes:

$$\mathbf{r} = \mathbf{H}\mathbf{x} - \mathbf{z}'. \quad (4.10)$$

And the optimization problem is now expressed as:

$$\text{Minimize } J = (\mathbf{H}\mathbf{x} - \mathbf{z}')^T \mathbf{W} (\mathbf{H}\mathbf{x} - \mathbf{z}'). \quad (4.11)$$

The optimal condition is when

$$0 = dJ/d\mathbf{x} = 2\mathbf{H}^T \mathbf{W} (\mathbf{H}\mathbf{x} - \mathbf{z}'). \quad (4.12)$$

The solution is:

$$\mathbf{x} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{z}'. \quad (4.13)$$

Upon substitution of the \mathbf{z}' vector, we generalize the solution as an iterative equation:

$$\mathbf{x}^{v+1} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{z}' = \mathbf{x}^v - (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} (h(\mathbf{x}^v) - \mathbf{z}). \quad (4.14)$$

After calculating the solution, we apply the chi-square test. The chi-square test provides a mathematical method of evaluating whether the measurements fit the system model. The procedure is as follows:

First, we compute the chi-square value as

$$\xi = \sum_i \left(\frac{h_i(\mathbf{x}) - z_i}{\sigma_i} \right)^2. \quad (4.15)$$

Then we apply the confidence level:

$$P = 1 - \Pr(\xi, \nu), \quad (4.16)$$

where ν is the degree of freedom, which is the difference between the number of measurements and states. If the confidence level remains 100%, it turns out that the measurements match the system model, and if it is 0, the system must contain bad data or hidden failures, and the bad data identification procedure is initiated. The state estimator will identify the bad data and remove them from the measurement set. At the end, the computed best estimate of the state of this section will be best for the given measurements.

The computed best estimate of this section and the network model are utilized to compute the best estimate of the bad data, if any, and the best estimate of missing data, if any:

$$\mathbf{z}_{\text{bad}} = \mathbf{Y}_{\text{eqz_bad}} \hat{\mathbf{x}} + \left\{ \begin{array}{c} \mathbf{M} \\ \hat{\mathbf{x}}^T \mathbf{F}_{\text{eqz_bad}}^i \hat{\mathbf{x}} \\ \mathbf{M} \end{array} \right\} - \mathbf{B}_{\text{eqz_bad}} + \boldsymbol{\eta} \quad (4.17)$$

$$\mathbf{z}_{\text{miss}} = \mathbf{Y}_{\text{eqz_miss}} \hat{\mathbf{x}} + \left\{ \begin{array}{c} \mathbf{M} \\ \hat{\mathbf{x}}^T \mathbf{F}_{\text{eqz_miss}}^i \hat{\mathbf{x}} \\ \mathbf{M} \end{array} \right\} - \mathbf{B}_{\text{eqz_miss}} + \boldsymbol{\eta} \quad (4.18)$$

where the model equations for the bad data and missing data are denoted with the subscript “bad” and “miss”, \hat{x} is the best estimate of this network.

If the confidence level remains high, then the measurements are consistent with the network model. In this case, the network model is validated, and the network model as well as the estimated operating conditions are transmitted to the distribution management system (DMS) for optimal control application.

Section 5: Optimal Power Flow Formation

One of the applications of DS-DQSE output is the full state feedback control of the distribution system. The DS-DQSE is able to continuously monitor the distribution network operating condition, validate the models, and deliver the information to the controller in less than two cycles. The accurate operating conditions as well as the validated models enable optimal use of distributed energy resources (DER) to achieve an objective such as voltage control. For this purpose on Optimal Power Flow is formulated using the validated model from the DS-DQSE as well as the operating conditions from the DS-DQSE. Note that the equality and inequality constraints of the OPF are constructed from the device-level and network-level models as described in Appendix D. This section introduces the details of the definition and formation of this OPF problem.

The OPF problem is formed using the quadratized model from DS-DQSE. By construction it is a quadratized OPF problem of the following mathematical form:

$$\begin{aligned}
 \text{Minimize: } J &= Y_{objx}^T \mathbf{x} + Y_{obju}^T \mathbf{u} + \mathbf{x}^T F_{objxx} \mathbf{x} + \mathbf{u}^T F_{objuu} \mathbf{u} + \mathbf{u}^T F_{objux} \mathbf{x} + C_{objc} \\
 \text{subject to: } 0 &= Y_{eqx} \mathbf{x} + Y_{equ} \mathbf{u} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T \langle F_{eqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{equ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{equx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} - B_{eq} - I \\
 B_{eq} &= -N_{eqx} \mathbf{x}(t-h) - N_{equ} \mathbf{u}(t-h) - M_{eq} I(t-h) - K_{eq} \\
 Y_{ineqx} \mathbf{x} + Y_{inequ} \mathbf{u} &+ \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T \langle F_{ineqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{inequ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{inequx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + C_{ineqc} \leq 0 \\
 \mathbf{u}_{\min} &\leq \mathbf{u} \leq \mathbf{u}_{\max}
 \end{aligned} \tag{5.1}$$

In this report, the objective is to improve the voltage profile across the network.

The formation of the OPF problem is achieved automatically by simply using the object-oriented SCAQCF distribution network model. The problem is stated as follows. Given the network model in SCAQCF syntax, define and form the various components in the OPF problem.

Section 5.1: Definition/Formation of Equality Constraints

This section introduces the definition and formation of equality constraints in the quadratized OPF problem. Since the equality constraints are obtained from the network model, they are also in the SCAQCF form and their general expression is:

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{u}) = Y_{eqx} \mathbf{x} + Y_{equ} \mathbf{u} + \left\{ \mathbf{x}^T \left\langle F_{eqx}^i \right\rangle \mathbf{x} \right\} + \left\{ \mathbf{u}^T \left\langle F_{equ}^i \right\rangle \mathbf{u} \right\} + \left\{ \mathbf{u}^T \left\langle F_{equx}^i \right\rangle \mathbf{x} \right\} - B_{eq} - I \quad (5.2)$$

$$B_{eq} = -N_{eqx} \mathbf{x}(t-h) - N_{equ} \mathbf{u}(t-h) - M_{eq} I(t-h) - K_{eq}$$

Connectivity: *TerminalNodeName*

Normalization Factor: StateNormFactor, ThroughNormFactor, ControlNormFactor

where

I : the through variables of the network model;

\mathbf{x} : external and internal state variables of the network model, $\mathbf{x} = [\mathbf{x}(t), \mathbf{x}(t_m)]$;

\mathbf{u} : control variables of the network model, $\mathbf{u} = [\mathbf{u}(t), \mathbf{u}(t_m)]$;

Y_{eqx} : matrix defining the linear part for state variables;

F_{eqx}^i : matrices defining the quadratic part for state variables;

Y_{equ} : matrix defining the linear part for control variables;

F_{equ}^i : matrices defining the quadratic part for control variables;

F_{equx}^i : matrices defining the quadratic part for the product of state and control variables;

B_{eq} : history dependent vector of the network model;

N_{eqx} : matrix defining the last integration step state variables part;

N_{equ} : matrix defining the last integration step control variables part;

M_{eq} : matrix defining the last integration step through variables part;

K_{eq} : constant vector of the network model.

TerminalNodeName : terminal names defining the connectivity of the network model;

StateNormFactor: Normalization Factors for the states;

ThroughNormFactor: Normalization Factors for the through and zero variables;

ControlNormFactor: Normalization Factors for the controls;

As shown in Figure 5.1, three components in the network SCAQCF model construct the equality constraints of the OPF problem. These three components are: (1) power flow equations, (2) network node names, and (3) state, through and control variables normalization factors. Notice that the formation procedure is object-oriented. In other words, given these three components as

inputs, we construct the equality constraints of the quadratized OPF problem as the output. The formation procedure first initializes the arrays defined for equality constraints, then copies the corresponding arrays from the network model to the equality constraints. The detailed object-oriented algorithm of equality constraints formation is illustrated in Appendix E.

$$\begin{aligned}
 & \begin{Bmatrix} \mathbf{I}(t) \\ 0 \\ 0 \\ \mathbf{I}(t_m) \\ 0 \\ 0 \end{Bmatrix} = Y_{net,eqx} \mathbf{x} + Y_{net,equ} \mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T \langle F_{net,eqx}^i \rangle \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \langle F_{net,equ}^i \rangle \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \langle F_{net,equx}^i \rangle \mathbf{x} \\ \vdots \end{Bmatrix} - B_{net,eq} \\
 & B_{net,eq} = -N_{net,eqx} \mathbf{x}(t-h) - N_{net,equ} \mathbf{u}(t-h) - M_{net,eq} \mathbf{I}(t-h) - K_{net,eq} \quad (1) \\
 & \mathbf{h}(\mathbf{x}, \mathbf{u}) = Y_{net,feqx} \mathbf{x} + Y_{net,fequ} \mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T \langle F_{net,feqx}^i \rangle \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \langle F_{net,fequ}^i \rangle \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \langle F_{net,fequx}^i \rangle \mathbf{x} \\ \vdots \end{Bmatrix} + C_{net,feq} \\
 & \text{Connectivity: } TerminalNodeName \quad (2) \\
 & \text{subject to: } \quad \mathbf{h}_{min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{max} \\
 & \quad \quad \quad \mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max} \\
 & \text{Normalization Factor: StateNormFactor, ThroughNormFactor, ControlNormFactor} \quad (3) \\
 & \text{Note: All the above variables are in metric system.}
 \end{aligned}$$

Figure 5.1: Three Components in Network SCAQCF Model for Constructing Equality Constraints

Section 5.2: Construction of Inequality Constraints at Network Level

This section introduces the definition and formation of inequality constraints in the quadratized OPF problem. Since the inequality constraints are obtained from the network model, they are also in the SCAQCF syntax and their general expression is:

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) = Y_{ineqx} \mathbf{x} + Y_{inequ} \mathbf{u} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{x}^T \langle F_{ineqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T \langle F_{inequ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T \langle F_{inequx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + C_{ineqc} \leq 0. \quad (5.3)$$

where

Y_{ineqx} : constraint matrix defining the linear part for state variables;

F_{ineqx} : constraint matrices defining the quadratic part for state variables;
 Y_{inequ} : constraint matrix defining the linear part for control variables;
 F_{inequ} : constraint matrices defining the quadratic part for control variables;
 F_{inequx} : constraint matrices defining the quadratic part for the product of state and control variables;
 C_{ineqc} : history dependent vectors for the inequality constraints.

As shown in Figure 5.2, two components in the network SCAQCF model construct the inequality constraints of the quadratized OPF problem. These two components are: (1) network functional constraint equations, and (2) upper bound and lower bound vectors of these functional constraints. Notice that the formation procedure is object-oriented. In other words, given these two components as inputs, we construct the inequality constraints of the quadratized OPF problem as the output. The formation procedure first initializes the arrays defined for inequality constraints, then transform the bilateral inequalities in the network model to the unilateral inequalities in the OPF problem. The detailed object-oriented algorithm of inequality constraints formation is illustrated in Appendix F.

$$\begin{Bmatrix} \mathbf{I}(t) \\ 0 \\ 0 \\ \mathbf{I}(t_m) \\ 0 \\ 0 \end{Bmatrix} = Y_{net,eqx} \mathbf{x} + Y_{net,equ} \mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T \langle F_{net,eqx}^i \rangle \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \langle F_{net,equ}^i \rangle \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \langle F_{net,eqix}^i \rangle \mathbf{x} \\ \vdots \end{Bmatrix} - B_{net,eq}$$

$$B_{net,eq} = -N_{net,eqx} \mathbf{x}(t-h) - N_{net,equ} \mathbf{u}(t-h) - M_{net,eq} \mathbf{I}(t-h) - K_{net,eq}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) = Y_{net,feqx} \mathbf{x} + Y_{net,fequ} \mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T \langle F_{net,feqx}^i \rangle \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \langle F_{net,fequ}^i \rangle \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \langle F_{net,fequx}^i \rangle \mathbf{x} \\ \vdots \end{Bmatrix} + C_{net,feq} \quad (1)$$

Connectivity: *TerminalNodeName*

$$\text{subject to: } \quad \mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{\max} \quad (2)$$

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$$

Normalization Factor: StateNormFactor, ThroughNormFactor, ControlNormFactor

Note: All the above variables are in metric system.

Figure 5.2: Two Components in Network SCAQCF Model for Constructing Inequality Constraints

Section 5.3: Construction of Control Constraints at Network Level

This section introduces the definition and formation of control constraints in the quadratized OPF problem. The control constraints in the OPF problem are directly obtained from the control constraints in the network model, and their general expression is:

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}. \quad (5.4)$$

where

\mathbf{u}_{\min} : lower bound vector for the control variables;

\mathbf{u}_{\max} : upper bound vector for the control variables.

The formation procedure is object-oriented, i.e., given the control constraints from the network model as the input, we compute the control constraints of the quadratized OPF problem as the output. The detailed object-oriented algorithm of control constraints formation is illustrated in Appendix G.

Section 5.4: Construction of Objective Function at Network Level

This section introduces the definition and formation of the objective function in the quadratized OPF problem. The objective function is defined as the minimization of the sum of the squares of the difference between the voltage magnitudes at selected nodes and the targeted voltage values. Since the voltage phasors are in Cartesian coordinates, the magnitude of a voltage phasor is not quadratized, but in a square root form. To solve this problem, we create a voltage magnitude model where the voltage magnitude is a state with “_MG” in its node name. Wherever it is desirable to control the voltage, a voltage magnitude model is placed at that node. Notice that the network model is formed by considering all the voltage magnitude models. Therefore, if a node name with “_MG” occurs in the network node name list, it is detected and automatically included. In this way, the objective function is expressed as:

$$\text{minimize: } J = \sum_{i \in \{\text{selected nodes/phases}\}} \left(\frac{V_{i,mag} - V_{i,target}}{\alpha_i V_{i,target}} \right)^2, \quad (5.5)$$

where $V_{i,mag}$ is a state of the network, which is the voltage magnitude of the selected nodes/phase to neutral voltage, $V_{i,target}$ is the corresponding targeted voltage value, and α_i is a user defined tolerance value (e.g., 4%). Note that $V_{i,mag}$ is identified by the corresponding node name with “_MG”, while $V_{i,target}$ and α_i are the parameters obtained from the corresponding voltage magnitude model.

The formation procedure is to expand the objective function and store the coefficients from different parts into corresponding arrays. The general quadratized format of the objective function is:

$$\text{Minimize: } J = Y_{objx}^T \mathbf{x} + Y_{obju}^T \mathbf{u} + \mathbf{x}^T F_{objxx} \mathbf{x} + \mathbf{u}^T F_{objuu} \mathbf{u} + \mathbf{u}^T F_{objux} \mathbf{x} + C_{objc}, \quad (5.6)$$

where

Y_{objx}^T : coefficients of the linear state variables in the objective function;

Y_{obju}^T : coefficients of the linear control variables in the objective function;

F_{objxx} : coefficients of quadratic state variables in the objective function;

F_{objuu} : coefficients of quadratic control variables in the objective function;

F_{objux} : coefficients of the production of state and control variables in the objective function;

C_{objc} : constant value in the objective function.

The objective function formation procedure is object-oriented, i.e., given the network node name list and its network index from the network model as inputs, we construct the arrays defined for the objective function as outputs. The detailed object-oriented algorithm is illustrated in Appendix H.

Section 6: Optimal Power Flow Solution Algorithm

This section introduces the optimal power flow solution algorithm after the OPF problem is defined and formed in Section 5. Given the defined optimization problem and the current operating point $(\mathbf{x}^0, \mathbf{u}^0)$, the algorithm first applies the co-state method to linearize the OPF problem so that the OPF problem becomes a linearized problem represented in terms of control variables only. The number of linearized operational constraints is only the inequality constraints that are close to their limits (modeled constraints). Operational constraints that are not near the limits do not need to be part of the OPF solution (un-modeled constraints). The control constraints are the physical upper and lower bounds of the control variables from those controllable devices. Then the algorithm computes the optimal values of the control variables using linear programming and solves the state variables by power flow equations. If the updated operating point violates the original quadratic modeled constraints, then the OPF solver modifies the b vector in the set of inequality constraints in linear programming, retrieves the operating point from the last iteration, and resolves the linearized optimization problem. If the updated operating point violates some of the un-modeled constraints, then the OPF solver adds these constraints, retrieves the operating point from the last iteration, linearizes the new constraints, and solves the linearized optimization problem again. The end result of the OPF solver is the optimal control output, i.e. the optimal values of the control variables. Figure 6.1 shows the flow chart of the algorithm.

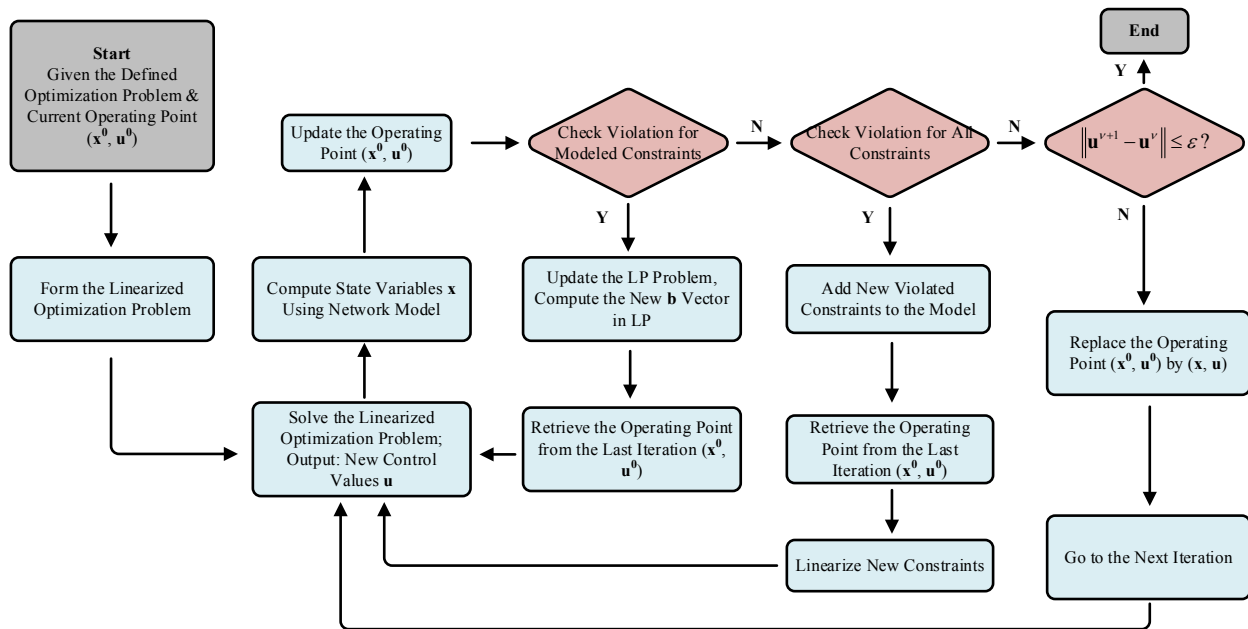


Figure 6.1: Flow Chart of the Algorithm

The proposed OPF solution algorithm is robust and highly efficient. The robustness means that the algorithm starts from a feasible but not optimal solution and moves the operating point in the feasible region while approaching the optimality. Therefore, the output of the algorithm is always a feasible solution. High efficiency means - less runtime compared to traditional solution methods for the OPF problem. The reasons are as follows: (1) The algorithm models the OPF problem as a quadratic problem for fast convergence; (2) The algorithm identifies the active constraints gradually and adds them to the modeled constraint set if needed. These features of the algorithm ensure that at each iteration, the dimension of the problem is the smallest possible for the specific distribution system section.

Section 6.1: Linearization

This subsection introduces the linearization of the quadratized OPF problem using the co-state method. The reason of using linearization techniques is: (1) The quadratized OPF problem consists of both state variables and control variables. To simplify the problem, we apply the co-state method so that the OPF problem becomes a linearized problem represented by only control variables; (2) After the linearization, the problem is transformed into a LP in standard form which is solve with a linear programming solver. A brief introduction of the linearization procedure is as follows. The detailed procedure is given in Appendix H.

Recall that the general expression of the OPF problem is:

$$\begin{aligned}
 \text{Minimize: } & J(\mathbf{x}, \mathbf{u}) = Y_{objx}^T \mathbf{x} + Y_{obju}^T \mathbf{u} + \mathbf{x}^T F_{objxx} \mathbf{x} + \mathbf{u}^T F_{objuu} \mathbf{u} + \mathbf{u}^T F_{objux} \mathbf{x} + C_{obj} \\
 \text{subject to: } & 0 = g(\mathbf{x}, \mathbf{u}) = Y_{eqx} \mathbf{x} + Y_{equ} \mathbf{u} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T \langle F_{eqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{equ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{equx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} - B_{eq} - I \\
 & B_{eq} = -N_{eqx} \mathbf{x}(t-h) - N_{equ} \mathbf{u}(t-h) - M_{eq} I(t-h) - K_{eq} \\
 & h(\mathbf{x}, \mathbf{u}) = Y_{ineqx} \mathbf{x} + Y_{inequ} \mathbf{u} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T \langle F_{ineqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{inequ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{inequx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + C_{ineqc} \leq 0 \\
 & \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}
 \end{aligned} \tag{6.1}$$

The formulated OPF problem is quadratic and in a standard format, so the linearization procedure is implemented as an object oriented program. Note that the objective function, inequality constraints, and control variables are the three components that will be linearized, while the equality constraints are taken into consideration during the linearization procedure. The algorithm is applied only to the modeled inequality constraints. Non-modeled inequality constraints need not to be linearized.

The final expression of the linearized objective function is:

$$J = \mathbf{c}^T \Delta \mathbf{u} + d_j, \quad (6.2)$$

where $\Delta \mathbf{u}$ is the increment of the control variable \mathbf{u} , $\Delta \mathbf{u} = \mathbf{u} - \mathbf{u}^0$, \mathbf{x}^0 and \mathbf{u}^0 are the current operating point, \mathbf{c} is the linear coefficient vector of $\Delta \mathbf{u}$,

$$\mathbf{c}^T = \frac{dJ(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} = \frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} - \hat{\mathbf{x}}_j \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}, \quad (6.3)$$

$\hat{\mathbf{x}}_j$ is the co-state vector regarding to the objective function,

$$\hat{\mathbf{x}}_j = - \frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \right)^{-1}, \quad (6.4)$$

and d_j is a constant value, $d_j = J(\mathbf{x}^0, \mathbf{u}^0)$.

The final expression of the linearized inequalities is:

$$\mathbf{a} \Delta \mathbf{u} + \mathbf{d} \leq 0, \quad (6.5)$$

Where \mathbf{a} is the linear coefficient matrix of $\Delta \mathbf{u}$,

$$\mathbf{a} = \frac{d\mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} = \frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} - \hat{\mathbf{x}}_h \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}, \quad (6.6)$$

$\hat{\mathbf{x}}_h$ is the co-state vector regarding to the inequalities,

$$\hat{\mathbf{x}}_h = - \frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \right)^{-1}, \quad (6.7)$$

and \mathbf{d} is the constant value vector, $\mathbf{d} = \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)$.

The constraints of the control variable are also changed to the constraint of the increment of control variable in the following way:

Substitute $\mathbf{u} = \mathbf{u}^0 + \Delta \mathbf{u}$ into the control variable constraint $\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$, we have:

$$\mathbf{u}_{\min} \leq \mathbf{u}^0 + \Delta \mathbf{u} \leq \mathbf{u}_{\max}. \quad (6.8)$$

Thus, the constraint of increment of control variable is:

$$\mathbf{u}_{\min} - \mathbf{u}^0 \leq \Delta \mathbf{u} \leq \mathbf{u}_{\max} - \mathbf{u}^0. \quad (6.9)$$

After the linearization of the quadratized OPF problem, we have the linearized problem with respect to $\Delta \mathbf{u}$ only, and its expression is:

$$\begin{aligned}
 \text{Minimize: } & J = \mathbf{c}^T \Delta \mathbf{u} + d_J \\
 \text{subject to: } & \mathbf{a} \Delta \mathbf{u} + \mathbf{d} \leq 0 \\
 & \mathbf{u}_{\min} - \mathbf{u}^0 \leq \Delta \mathbf{u} \leq \mathbf{u}_{\max} - \mathbf{u}^0
 \end{aligned} \tag{6.10}$$

Section 6.2: Solution of the Linearized Problem

This subsection introduces the procedure to solve the linearized optimization problem defined by equation set (6.10) by a standard linear programming (i.e., simplex method) solver. Recall that the linearized problem consists of inequality constraints, and all its variables are free variables. Simplex method solvers require that all variables be non-negative as shown in equation set (6.11).

$$\begin{aligned}
 \text{Minimize: } & \mathbf{c}^T \mathbf{x} \\
 \text{subject to: } & \mathbf{A} \mathbf{x} = \mathbf{B} \\
 & \mathbf{x} \geq 0
 \end{aligned} \tag{6.11}$$

In order to (1) transform the inequality constraints in the linearized problem to the equality constraints in the standard form, and (2) transform the free variables in the linearized problem to the non-negative variables in the standard form, we need to introduce non-negative variables to the linearized optimization problem.

First, we introduce non-negative variables s_i^+ and s_i^- into the objective function:

$$\forall i, \Delta u_i = s_i^+ - s_i^-, \tag{6.12}$$

where $s_i^+ \geq 0$ and $s_i^- \geq 0$.

Denote $\mathbf{s}^+ = [s_1^+ \quad \dots \quad s_n^+]^T$, and $\mathbf{s}^- = [s_1^- \quad \dots \quad s_n^-]^T$, where n is the total number of the control variables.

Then, the objective function is in the following form:

$$\text{Minimize: } J = \begin{bmatrix} \mathbf{c}^T & -\mathbf{c}^T \end{bmatrix} \begin{bmatrix} \mathbf{s}^+ \\ \mathbf{s}^- \end{bmatrix}. \tag{6.13}$$

The constraints are then changed into the standard form. For each inequality constraint $\mathbf{a}_{irow} \Delta u_i + d_i \leq 0$, a non-negative variable y_i is introduced. And the inequality constraint is transformed into the equality constraint:

$$\mathbf{a}_{irow} (s_i^+ - s_i^-) + y_i + d_i = 0, \tag{6.14}$$

where $y_i \geq 0$.

For each control variable constraint $u_{\min}^i - u_0^i \leq s_i^+ - s_i^- \leq u_{\max}^i - u_0^i$, non-negative variables p_i and q_i are introduced, so that $s_i^+ - s_i^- \geq u_{\min}^i - u_0^i$ is transformed to

$$s_i^+ - s_i^- - p_i - u_{\min}^i + u_0^i = 0. \quad (6.15)$$

And $s_i^+ - s_i^- \leq u_{\max}^i - u_0^i$ is transformed to

$$s_i^+ - s_i^- + q_i - u_{\max}^i + u_0^i = 0. \quad (6.16)$$

Thus, the inequality constraints are changed to:

$$\begin{aligned} [\mathbf{a} \quad -\mathbf{a}] \begin{bmatrix} \mathbf{s}^+ \\ \mathbf{s}^- \end{bmatrix} + \mathbf{y} + \mathbf{d} &= 0 \\ [I \quad -I] \begin{bmatrix} \mathbf{s}^+ \\ \mathbf{s}^- \end{bmatrix} - \mathbf{p} - \mathbf{u}_{\min} + \mathbf{u}^0 &= 0, \\ [I \quad -I] \begin{bmatrix} \mathbf{s}^+ \\ \mathbf{s}^- \end{bmatrix} + \mathbf{q} - \mathbf{u}_{\max} + \mathbf{u}^0 &= 0 \\ \mathbf{s}^+, \mathbf{s}^-, \mathbf{y}, \mathbf{p}, \mathbf{q} &\geq 0 \end{aligned} \quad (6.17)$$

And the problem is now in the following standard form:

$$\begin{aligned} \text{Minimize: } J &= C^T \mathbf{z} + d \\ \text{subject to: } A\mathbf{z} &= B, \\ \mathbf{z} &\geq 0 \end{aligned} \quad (6.18)$$

$$\text{where } C = \begin{bmatrix} \mathbf{c} \\ -\mathbf{c} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{z} = \begin{bmatrix} \mathbf{s}^+ \\ \mathbf{s}^- \\ \mathbf{y} \\ \mathbf{p} \\ \mathbf{q} \end{bmatrix}, A = \begin{bmatrix} \mathbf{a} & -\mathbf{a} & I & 0 & 0 \\ I & -I & 0 & -I & 0 \\ I & -I & 0 & 0 & I \end{bmatrix}, \text{ and } B = \begin{bmatrix} -\mathbf{d} \\ \mathbf{u}_{\min} - \mathbf{u}^0 \\ \mathbf{u}_{\max} - \mathbf{u}^0 \end{bmatrix}.$$

The solution of the linearized optimization problem is obtained by a standard linear programming solver (i.e., simplex method). The solution provides the variables $\mathbf{s}^+, \mathbf{s}^-$ and $\Delta \mathbf{u}$. The updated optimal values of the control variables are: $\mathbf{u} = \mathbf{u}^0 + \Delta \mathbf{u}$.

Section 6.3: Equality Equations Solution (Power Flow Problem)

Once the new control values are obtained, the network states need to be updated accordingly. As the state and control variables obey the power flow equations $g(\mathbf{x}, \mathbf{u})=0$, we use power flow equations to solve for the updated states by substituting the new control values into the control variables. The details are illustrated below.

Recall that the power flow equations $g(\mathbf{x}, \mathbf{u})=0$ is:

$$0 = g(\mathbf{x}, \mathbf{u}) = 0 = Y_{eqx} \mathbf{x} + Y_{equ} \mathbf{u} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T \langle F_{eqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{equ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T \langle F_{equx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{array} \right\} - B_{eq} - I \quad (6.19)$$

$$B_{eq} = -N_{eqx} \mathbf{x}(t-h) - N_{equ} \mathbf{u}(t-h) - M_{eq} I(t-h) - K_{eq}$$

The updated states are solved by the Newton-Raphson method. And the detailed procedure is as follows.

- (1) Let $\nu=0$ and $\mathbf{x} = \mathbf{x}^\nu$, where ν is the iteration number to obtain the states \mathbf{x} in the Newton-Raphson method.
- (2) Substitute \mathbf{x}^ν and \mathbf{u} into the power flow equation $g(\mathbf{x}^\nu, \mathbf{u})$, and compute $g(\mathbf{x}^\nu, \mathbf{u})$. If $\|g(\mathbf{x}^\nu, \mathbf{u})\| \leq \varepsilon$, \mathbf{x}^ν is the solution and the procedure is terminated, where ε is a user-defined small value that is used to determine whether the solution is converged. Otherwise, go to step (3).
- (3) Compute the Jacobian matrix: $\frac{\partial g(\mathbf{x}^\nu, \mathbf{u})}{\partial \mathbf{x}}$. The Jacobian matrix can be easily achieved as illustrated in Section 6.1.
- (4) Compute $\mathbf{x}^{\nu+1} = \mathbf{x}^\nu - \left(\frac{\partial g(\mathbf{x}^\nu, \mathbf{u})}{\partial \mathbf{x}} \right)^{-1} g(\mathbf{x}^\nu, \mathbf{u})$.
- (5) $\nu = \nu + 1$. If $\nu \leq \nu_{\max}$, go to step (2); otherwise, return nonconvergence. (ν_{\max} is the user-defined maximum number of iterations allowed to compute the states \mathbf{x} , and ν_{\max} is set to be 15 in this algorithm.)

The new operating point (\mathbf{x} and \mathbf{u}) is formed from the above computed values.

Section 6.4: Iterative Linearization/Solution Method

Since the new operating point is computed from the linearized optimization problem, it may overshoot and violate some of the original quadratic inequality constraints because of linearization errors. Some of these modeled operational constraints may be out of their bounds, especially for those constraints reaching upper bounds in the linearized optimization problem. Therefore, we need to check whether the active constraints $\mathbf{h}(\mathbf{x}, \mathbf{u}) \leq 0$ still hold at the new operating point.

If any modeled constraint is violated, the algorithm updates its corresponding constant item b in the linearized optimization problem, retrieves the previous solution, and solves the updated linearization problem and the power flow problem again.

Note that for each linearized inequality constraint, we have: $h(\mathbf{x}^0, \mathbf{u}^0) + \frac{dh(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} \Delta \mathbf{u} \leq 0$, and

the constant item b is: $\frac{dh(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} \Delta \mathbf{u} \leq -h(\mathbf{x}^0, \mathbf{u}^0) = b$.

The details of modifying the constant item b are as follows.

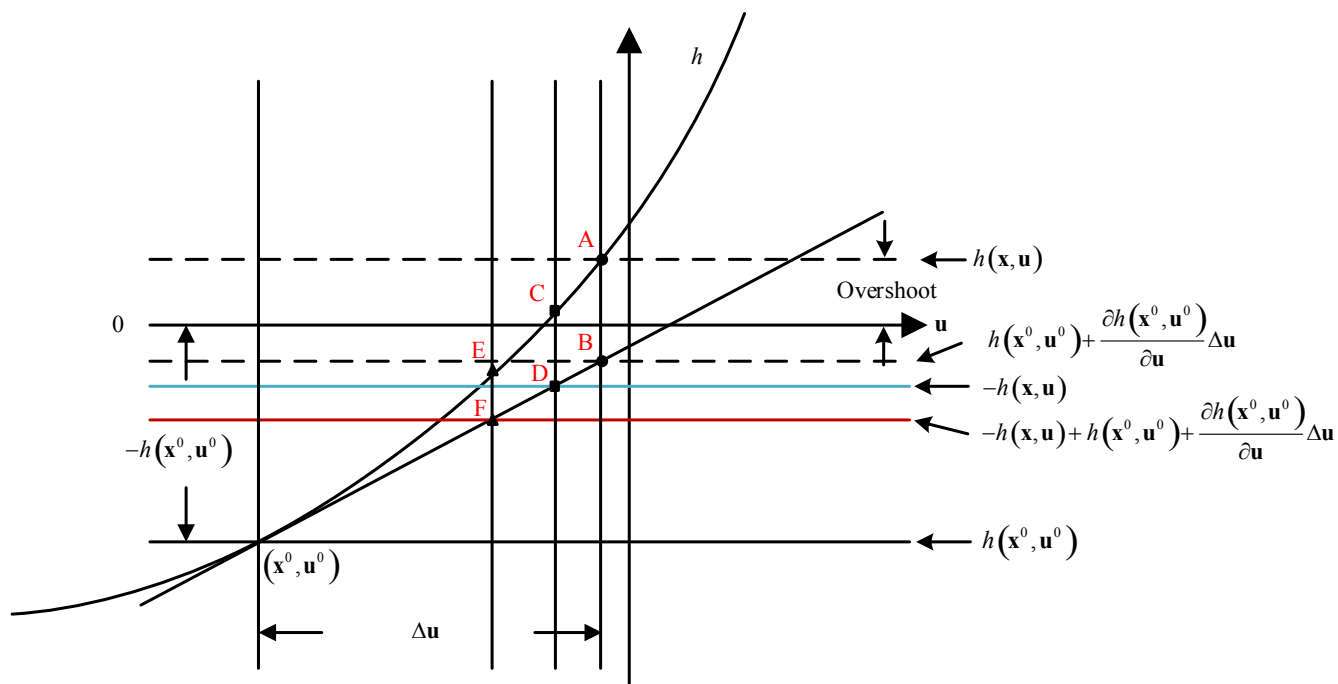


Figure 6.2: Linearization Update Method

As shown in Figure 6.2, the inequality constraint violation is caused by the linearization error of the control variable \mathbf{u} . Firstly, the LP result is $\Delta \mathbf{u}$, and point B is the operating point for the

linearized problem and point A is the operating point for the nonlinear problem. Although point B still does not violate the constraint, point A is above the upper bound. And the overshoot is $h(\mathbf{x}, \mathbf{u}) - 0 = h(\mathbf{x}, \mathbf{u})$. To solve this problem, the overshoot $h(\mathbf{x}, \mathbf{u})$ is subtracted from the upper bound. The solution of the linearized optimization problem moves from point B to point D in the figure, and the solution point for the nonlinear problem moves from point A to point C. The constraints will then be satisfied in most cases. However, in the situation as shown in Figure 6.2, the point C still violates the constraint. Therefore, $-h(\mathbf{x}^0, \mathbf{u}^0) - \frac{\partial h(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} \Delta \mathbf{u}$ shall also be subtracted from the constant item b, and the new operating points move from square points to the triangular points (point E and F). In this way, both the operating points of the linearized problem and the nonlinear problem satisfy the constraints. Thus, the constant item b is modified as:

$$-h(\mathbf{x}^0, \mathbf{u}^0), \text{ if } h(\mathbf{x}, \mathbf{u}) \leq 0 \text{ is not violated or}$$

$$-h(\mathbf{x}, \mathbf{u}) + \frac{\partial h(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} \Delta \mathbf{u}, \text{ if } h(\mathbf{x}, \mathbf{u}) \leq 0 \text{ is violated.}$$

Section 6.5: Determining Convergence or Addition of New Constraints

Once the modeled constraints are all satisfied, the algorithm will check all the remaining un-modeled constraints. Since the linearized problem does not include all the operating constraints, the new operating point may not satisfy some of them.

If the updated operating point satisfies all the un-modeled constraints, the algorithm has converged. The current iterate operating point (\mathbf{x} and \mathbf{u}) is the optimal operating point of the system. Otherwise, i.e. if one or more un-modeled operating constraints is violated, the algorithm stores the current operating point and proceeds to the next iteration.

In this case, the algorithm adds the un-modeled violated constraints into the OPF, linearizes the newly added constraints, and solves the updated linearized optimization problem and power flow equations. This is achieved by the following two steps.

Step 1: Check violations for all constraints and add new violated constraints to the linearized optimization problem model.

This step checks all the operating constraints defined in the OPF problem. If any un-modeled constraint is not satisfied, the algorithm adds this constraint and continues to check whether the violation exists in the rest of the operating constraints.

Step 2: Linearize the new constraints and retrieve the previous operating point.

This step is to linearize the new constraints and add the new linearized constraints to the linearized optimization problem. The OPF performs another iteration considering all the modeled

constraints (including the newly added ones). The linearization technique is introduced in Section 6.1. The procedure of solving the updated operating point is introduced from Section 6.2 to Section 6.4.

Section 7: Description of Example System

This section describes the proposed distribution system. Figure 7.1 shows the proposed distribution system consisting of three substations and two feeders. Feeder A contains three sections while feeder B contains two sections. Each section has several IEDs and a DS-DQSE, while a master state estimator monitors the whole system and processes the output data from the local state estimator in each section. The details of each section are given in the following paragraphs.

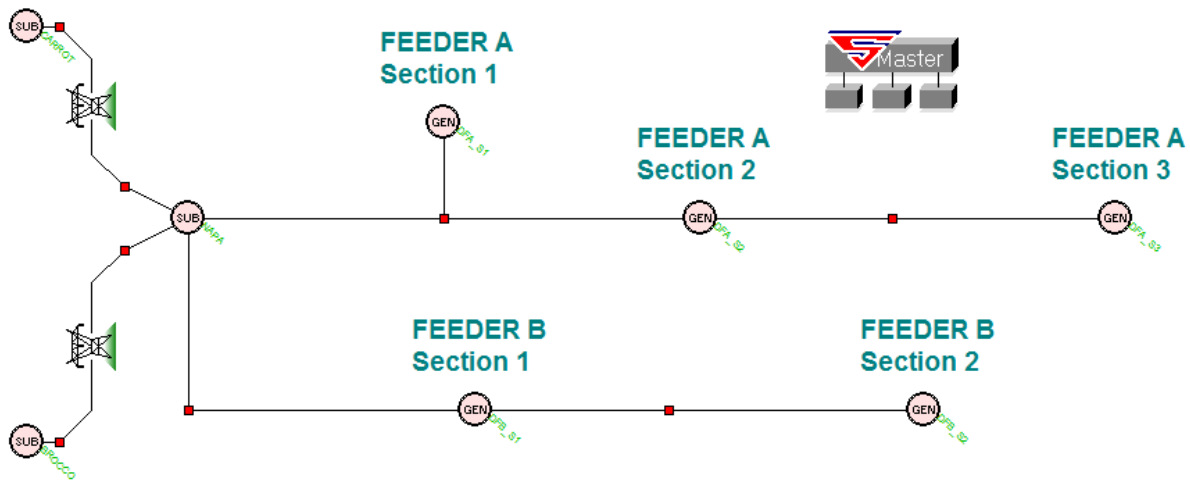


Figure 7.1: Proposed Distribution System

Figure 7.2 shows the feeder A, section 1 that consists of four distribution lines (13.8kV), one capacitor bank, one delta-wye transformer, and five loads. Four IEDs are installed in this section. IED_1 monitors the three-phase voltage and current phasors at high voltage side of the transformer (B12), IED_2 collects the data from the capacitor bank (B25), IED_3 and IED_4 measure the three-phase voltage and current phasors of the distribution lines (B13 and B16). Besides, one local state estimator is installed to collect the data from all IEDs and performs quasi-dynamic state estimation.

DIST Feeder A, Section 1

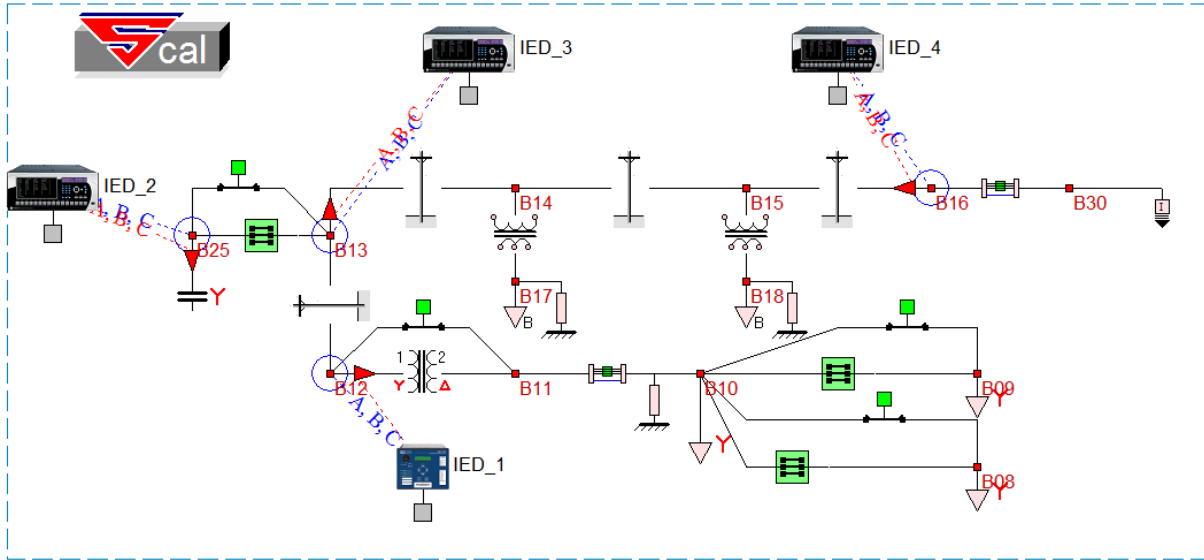


Figure 7.2: Distribution Feeder A, Section 1

Figure 7.3 shows the distribution feeder A, section 2 containing two distribution lines (13.8) kV, two reclosers, one capacitor bank, three single-phase lines, two single-phase transformers with secondary center-tap, two loads (residential loads), one battery, one converter and one three-phase transformer. Six IEDs are installed in this section: IED_1 and IED_3 monitor three-phase voltage and current phasors at the breakers (B111 and B205); IED_2 collects the data at the transformer (B200); IED_4 and IED_5 measure the single-phase voltage and current phasors at B201 and B206. IED_6 collects the data from the capacitor bank (B205). Besides, there is one local state estimator collecting the data from all IEDs and performing quasi-dynamic state estimation in this section.

DIST Feeder A, Section 2

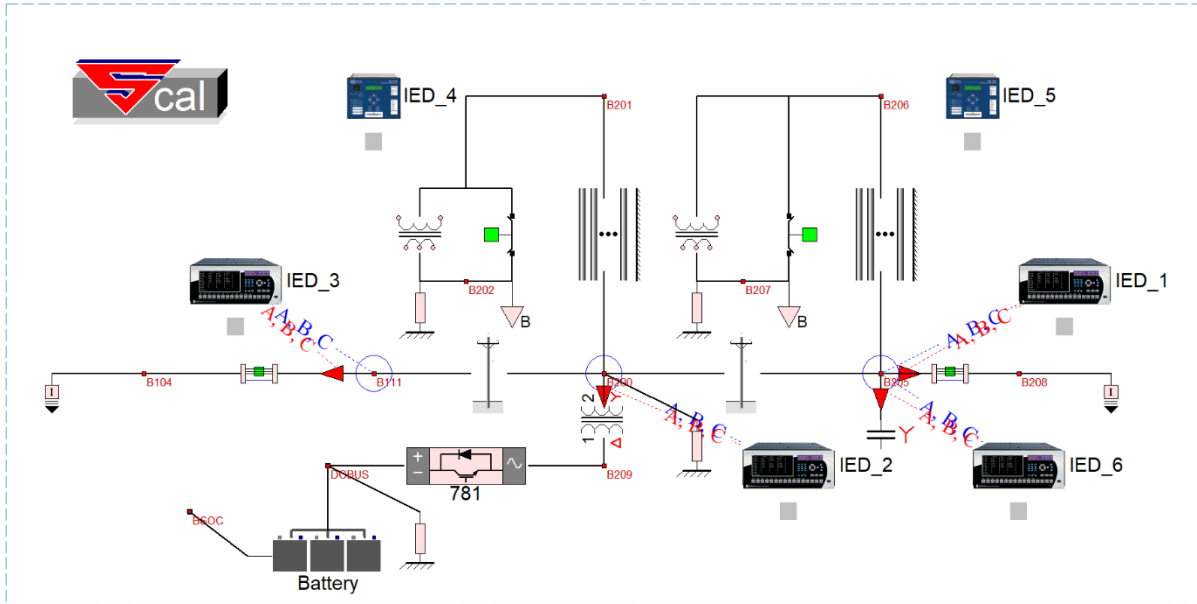


Figure 7.3: Distribution Feeder A, Section 2

Figure 7.4 shows the distribution feeder A, section 3 consisting of one distribution line (13.8kV), one recloser, one capacitor bank, one single-phase line, one single-phase transformer with secondary center-tap and one load (residential load). Three IEDs are installed in this section: IED_1 monitors the three-phase voltage and current phasors at a breaker (Bus209), IED_2 collects the data from the capacitor bank (Bus210), and IED_3 measures the single-phase voltage and current phasor at Bus211. As same as sections 1 and 2 of feeder A, there is one local state estimator in charge of collecting the data from all IEDs and performing quasi-dynamic state estimation in this section.

DIST Feeder A, Section 3

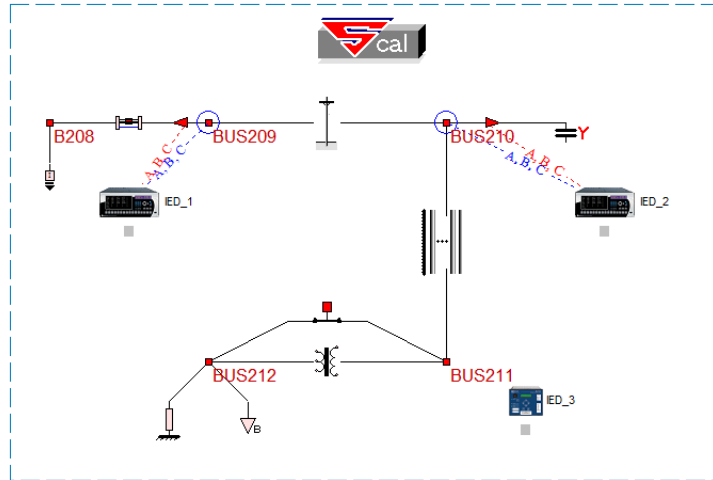


Figure 7.4: Distribution System, Feeder A, Section 3

Figure 7.5 shows the distribution feeder B, section 1 that contains three distribution lines (13.8kV), two reclosers, one capacitor bank, one delta-wye transformer and one load (induction motor, industrial load). Four IEDs are installed in this section: IED_1 and IED_3 monitor three-phase voltage and current phasors at breakers (Bus301 and Bus305); IED_2 collects the data from the capacitor bank (Bus302); IED_4 monitors three-phase voltage and current phasors at Bus303. Besides, one local state estimator collects the data from all IEDs and performs quasi-dynamic state estimation in this section.

DIST Feeder B, Section 1

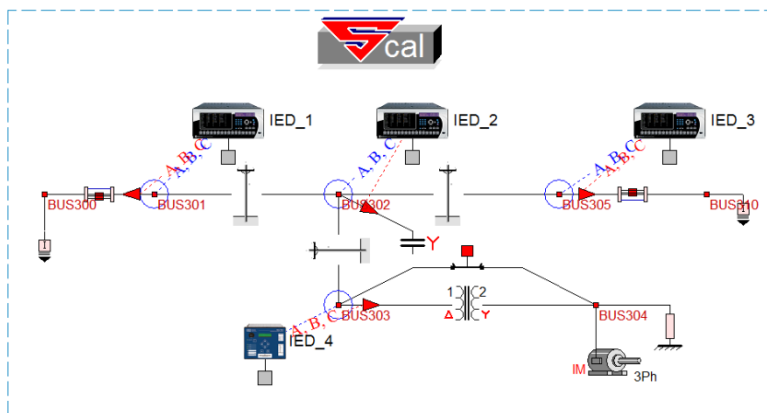


Figure 7.5: Distribution System, Feeder B, Section 1

Figure 7.6 shows the distribution feeder B, section 2 containing one distribution line (13.8kV), one recloser, one capacitor bank, one single-phase line, one single-phase transformer with secondary center-tap and one load (residential load). Three IEDs are installed in this section:

IED_1 monitors the breaker at Bus400, IED_2 collects the data from the capacitor bank (Bus401), and IED_3 measures the single-phase voltage and current phasors at Bus402. Besides, one local state estimator collects the data from all IEDs and performs quasi-dynamic state estimation in this section.

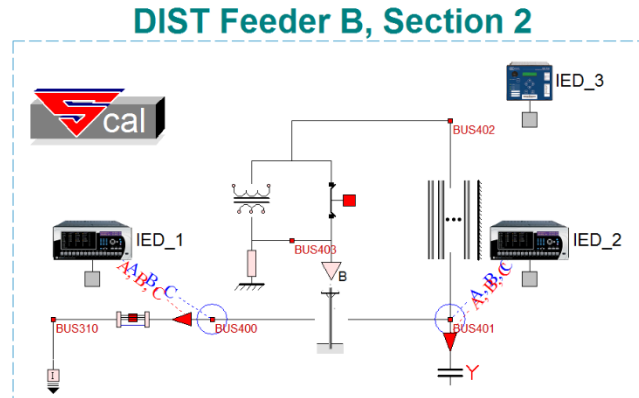


Figure 7.6: Distribution System, Feeder B, Section 2

In this report, Feeder A, Section 1 is investigated and analyzed. The local state estimator in Feeder A, Section 1 runs a 60-second event to test its performance. The figure of Feeder A, Section 1 is shown in Figure 7.2. The parameters of this section are as follows.

Distribution line 1 (B12 to B13), distribution line 2 (B13 to B14), distribution line 3 (B14 to B15) and distribution line 4 (B15 to B16) are 0.5 miles, 0.2 miles, 0.2 miles, and 0.3 miles, respectively, and they are all operating at 13.8 kV.

The three-phase delta-wye transformer (13.8kV to 0.48kV) is rated at 36.0 MVA with 0.002 p.u. winding resistance and 0.05 p.u. leakage reactance.

A capacitor bank is located at B25 for reactive power compensation. The rated voltage is 13.8 kV and the rated reactive power is 600kVAR.

Three-phase loads are located at B10, B09, and B08, respectively. These loads are considered as residential loads with 0.48kV rated voltage. The load at B10 is rated at 1600kW real power and 400kVAR reactive power, while the loads at B09 and B08 are with 800kW real power and 200kVAR reactive power consumption.

Besides, there are two loads with the same ratings (0.24 kV, 10kW, 3kVAR) at B14 and B15, respectively.

Four IEDs are installed in this section. IED_1 is SEL-734 and IED_2, IED_3, IED_4 are GE-D60. IED_1 measures three-phase voltage phasors at B12 and three-phase current phasors from B12 to B11. IED_2 at B25 measures three-phase voltage and current phasors for the capacitor bank. IED_3 measures three-phase voltage phasors at B13 and three-phase current phasors from B13 to B14. IED_4 at B16 measures three-phase voltage phasors at B16 and three-phase current phasors from B16 to B15. The meter scales for the voltage and the current are 13.8kV and 400A, respectively.

Section 8: Example DS-DQSE Results

This section presents the implementation of DS-DQSE.

Section 8.1: Measurement Creation

The quasi-dynamic state estimation is performed in Feeder A, Section 1. Note that this section contains 13 devices and 116 states, but only 24 phasor measurements (i.e. 48 measurements if the phasor measurement is partitioned into magnitude and phase angle measurement) are available. Therefore, the section is unobservable because of limited relays. In order to make this section observable, derived measurements, pseudo-measurements and virtual measurements are proposed, and these measurements are created based on the system topology and actual measurements from relays.

In general, the measurements in a specific section are classified into four types: (a) Actual Measurements: measurements that come from actual measurement channels, i.e. any measurements from any IEDs (relays, meters, FDR, PMUs, etc.); (b) Derived Measurements: measurements derived from actual measurements based on topology; (c) Pseudo Measurements: not directly measured, quantities represented for which their value is approximately known, such as missing phase measurements, neutral/shield voltage measurements, neutral currents, etc. (d) Virtual Measurements: mathematical quantities defined by physical laws, such as Kirchhoff's current law, model equations, etc.

The derived measurements and virtual measurements are easily created. For example, in Figure 8.1, the current from B12 to B13 is approximately set as the opposite direction current from B12 to B11 as:

$$I_{B12_B13,abc}^{\%} = -I_{B12_B11,abc}^{\%} \quad (8.1)$$

And the current from B13 to B12 is computed by applying Kirchhoff Current Law:

$$I_{B13_B12,abc}^{\%} = -\left(I_{B13_B14,abc}^{\%} + I_{B25,abc}^{\%}\right) \quad (8.2)$$

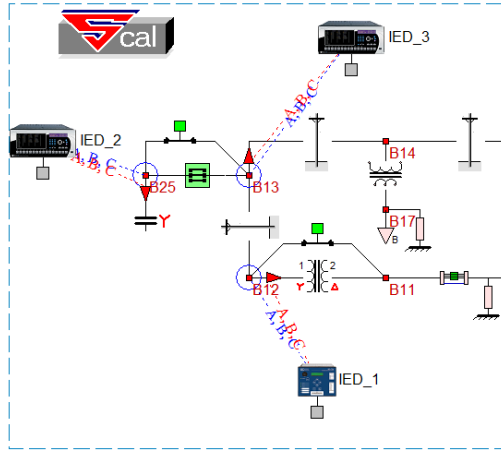


Figure 8.1: An example for derived measurements in Feeder A, Section 1

The pseudo-measurements in this section are created as follows:

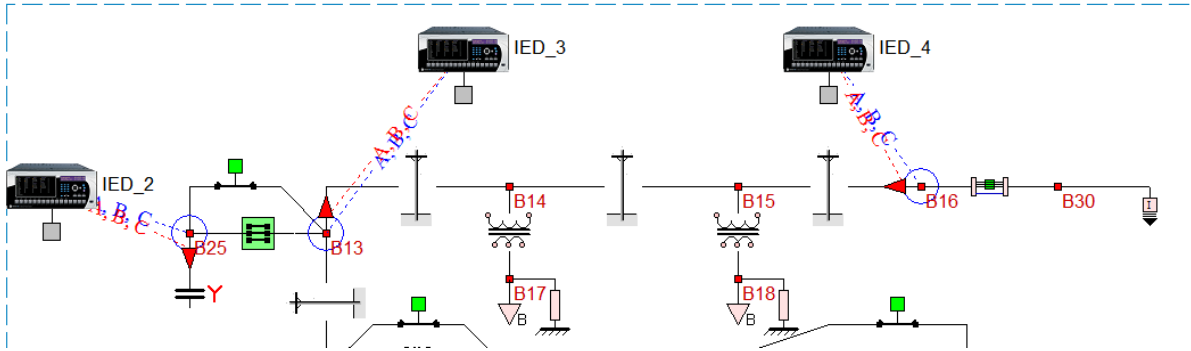


Figure 8.2 An example for pseudo-measurements in Feeder A, Section 1

Figure 8.2 shows three distribution lines from B13 to B16. The length of these three lines are 0.2, 0.2 and 0.3 miles, respectively. Two loads at Bus17 and Bus18 are identical single-phase loads at phase A. The voltage and current measurements are only available at B13 and B16. The expected voltage measurements at B14 and B15 are approximately computed as:

$$V_{B14,abc}^{\%} = \frac{0.5V_{B13,abc}^{\%} + 0.2V_{B16,abc}^{\%}}{0.7} \quad (8.3)$$

$$V_{B15,abc}^{\%} = \frac{0.3V_{B13,abc}^{\%} + 0.4V_{B16,abc}^{\%}}{0.7} \quad (8.4)$$

The expected current measurements from B14 to B17 and from B15 to B18 are computed as:

$$I_{B14_B17,abc}^{\%} = 0.5 (I_{B13,abc}^{\%} + I_{B16,abc}^{\%}) \quad (8.5)$$

$$I_{B15_B18,abc}^{\%} = 0.5 (I_{B13,abc}^{\%} + I_{B16,abc}^{\%}) \quad (8.6)$$

The expected current measurements from B14 to B13 and from B14 to B15 are:

$$I_{B14_B13,abc}^{\%} = -I_{B13_B14,abc}^{\%} \quad (8.7)$$

$$I_{B14_B15,abc}^{\%} = I_{B13_B14,abc}^{\%} - I_{B14_B17,abc}^{\%} = 0.5I_{B13_B14,abc}^{\%} - 0.5I_{B16_B15,abc}^{\%} \quad (8.8)$$

The expected current measurements from Bus105 to Bus104 and from Bus105 to Bus106 are:

$$I_{B15_B14,abc}^{\%} = -I_{B14_B15,abc}^{\%} = -0.5I_{B13_B14,abc}^{\%} + 0.5I_{B16_B15,abc}^{\%} \quad (8.9)$$

$$I_{B15_B16,abc}^{\%} = I_{B14_B15,abc}^{\%} - I_{B15_B18,abc}^{\%} = -I_{B16_B15,abc}^{\%} \quad (8.10)$$

Figure 8.3 shows a part of feeder A, section 1. A three-phase transformer is between B12 to B11. Three-phase loads are located at B10, B09, and B08. The relay is installed at B12, and therefore, the voltage and current measurements at B11, B10, B09 and B08 are unobservable. Since the voltage at the transformer Y side are highly dependent on the amount of the load, the pseudo measurement for the voltage at the delta side is not accurate. Therefore, only the currents at the low voltage side are considered. The expected current measurements from B11 to B12 are computed as:

$$\begin{aligned} I_{B11a}^{\%} &= (I_{B12ab}^{\%} - I_{B12ca}^{\%}) \times TXM_ratio \\ I_{B11b}^{\%} &= (I_{B12bc}^{\%} - I_{B12ab}^{\%}) \times TXM_ratio \\ I_{B11c}^{\%} &= (I_{B12ca}^{\%} - I_{B12bc}^{\%}) \times TXM_ratio \\ I_{B11n}^{\%} &= -(I_{B11a}^{\%} + I_{B11b}^{\%} + I_{B11c}^{\%}) \end{aligned} \quad (8.11)$$

As the rated power of the load is known, the expected current measurements at each load are approximately computed by the rated power ratio between these loads. In this section, if all the three loads are connected to the grid, the expected current measurements at B10, B09 and B08 are:

$$\begin{aligned} I_{B10abc}^{\%} &= -0.5I_{B11abc}^{\%} \\ I_{B09abc}^{\%} &= 0.5I_{B10abc}^{\%} \\ I_{B08abc}^{\%} &= 0.5I_{B10abc}^{\%} \end{aligned} \quad (8.12)$$

If only the loads at B10 and B09 are connected to the grid, the expected current measurements at B10, B09, and B08 are:

$$\begin{aligned} I_{B10abc}^{\%} &= -\frac{2}{3}I_{B11abc}^{\%} \\ I_{B09abc}^{\%} &= \frac{1}{2}I_{B10abc}^{\%} \\ I_{B08abc}^{\%} &= 0 \end{aligned} \quad (8.13)$$

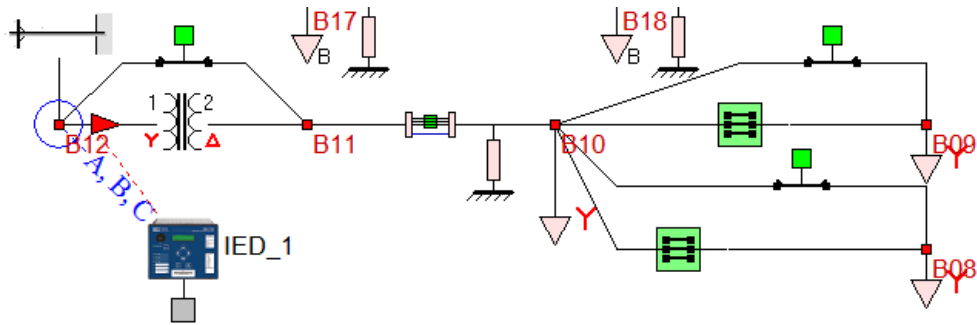


Figure 8.3: An example for pseudo-measurements in Feeder A, Section 1

All these measurements are treated as pseudo-measurements with relatively high standard deviation, e.g. 0.1 p.u. In addition, the standard deviation of actual measurements and virtual measurements are usually set as 0.01 p.u. and 0.001 p.u. respectively.

After adding pseudo-measurements, derived measurements, and virtual measurements, the system is observable. In this section, in addition to the 48 actual measurements, 70 pseudo-measurements, 12 derived measurements, and 6 virtual measurements (136 measurements in total) are available. The redundancy is $(136-116)/116=17.2\%$.

Section 8.2: QSE Implementation

The user interface of the state estimator is shown in Figure 8.4. The design of the state estimator supports the connection of the state estimator to a section bus or a data concentrator such as a PDC (phasor data concentrator), etc. In this way, the implementation in any section is straightforward and requires relatively short time.

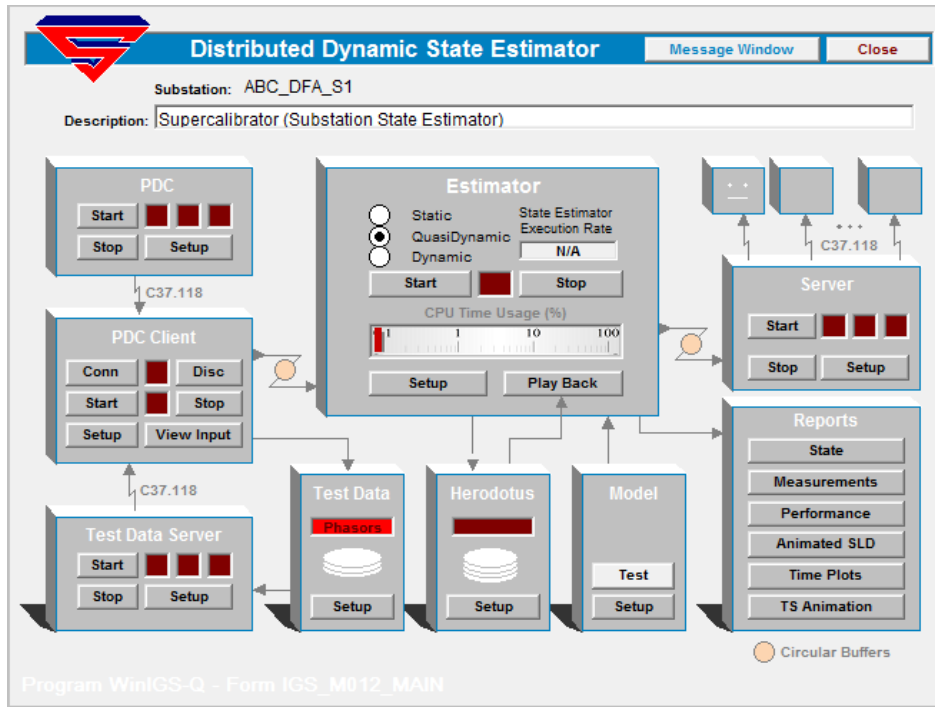


Figure 8.4: User Interface of State Estimator

The PDC client connects either to a PDC (for real time data) or to the test data server (for simulation experiments) and receives a C37.118 synchro-phasor data stream that may also be mixed with data from standard relays. The test data server serves the purpose of numerical experimentation with the quasi-state estimation where a C37.118 data stream is created that is served to the PDC client. The user can define the data source to be a synchro-phasor, phasor data, or time domain file that has been saved upon simulation of the system. The measurement data is streamed to the test data server before the state estimation, and the interface of the synchro-phasor test server and PDC client are shown in Figure 8.5 and Figure 8.6, respectively. Note that a PDC Client Data Frame Report is automatically generated to display the input measurements in details as shown in Figure 8.7.

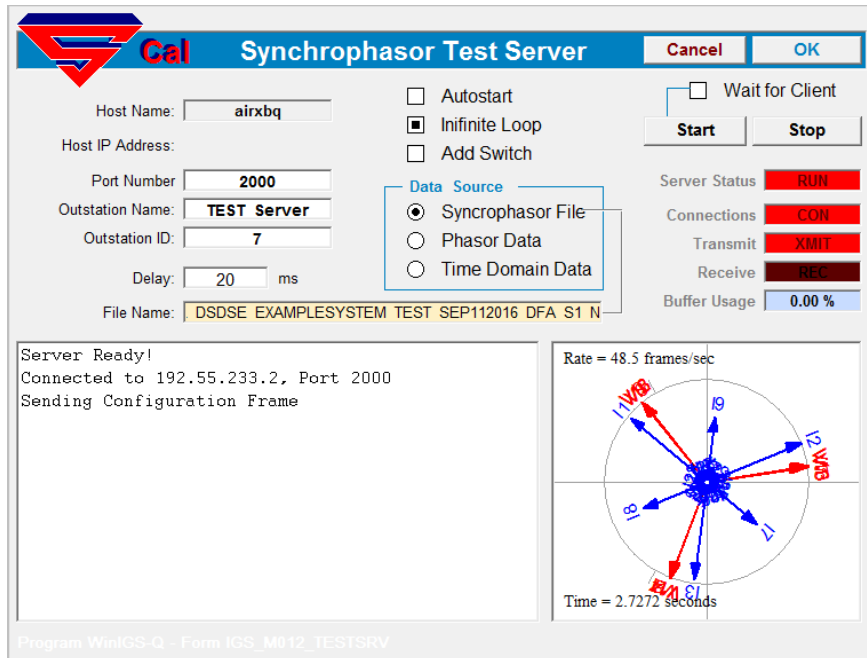


Figure 8.5: Test Data Server Interface

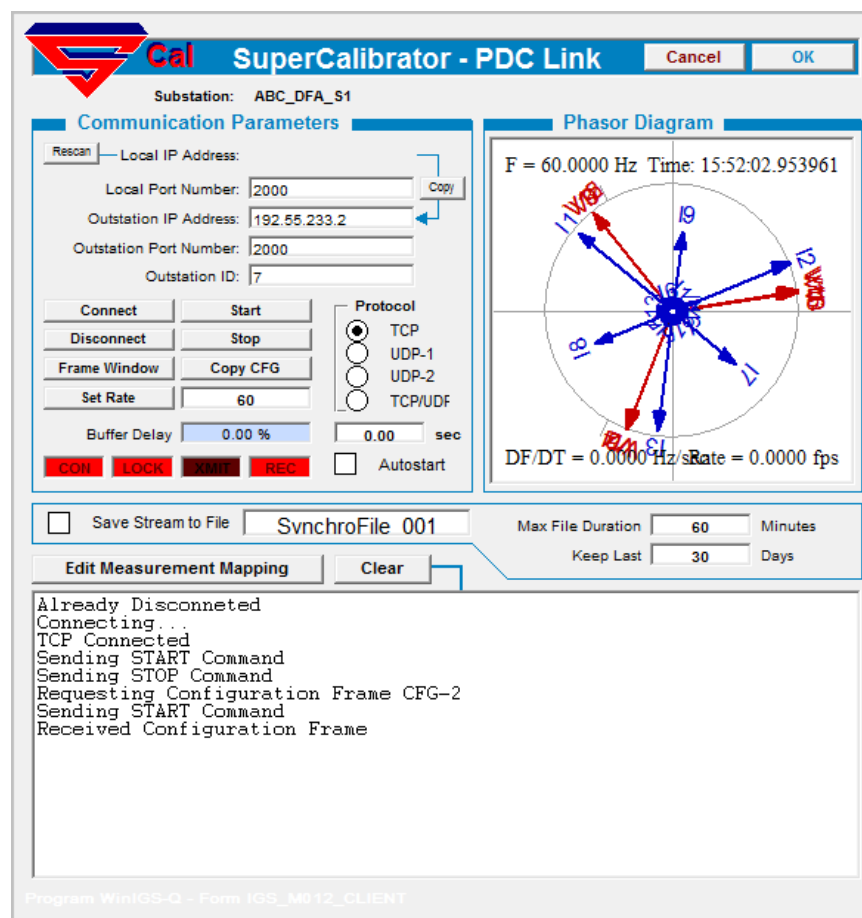


Figure 8.6: PDC Client Interface

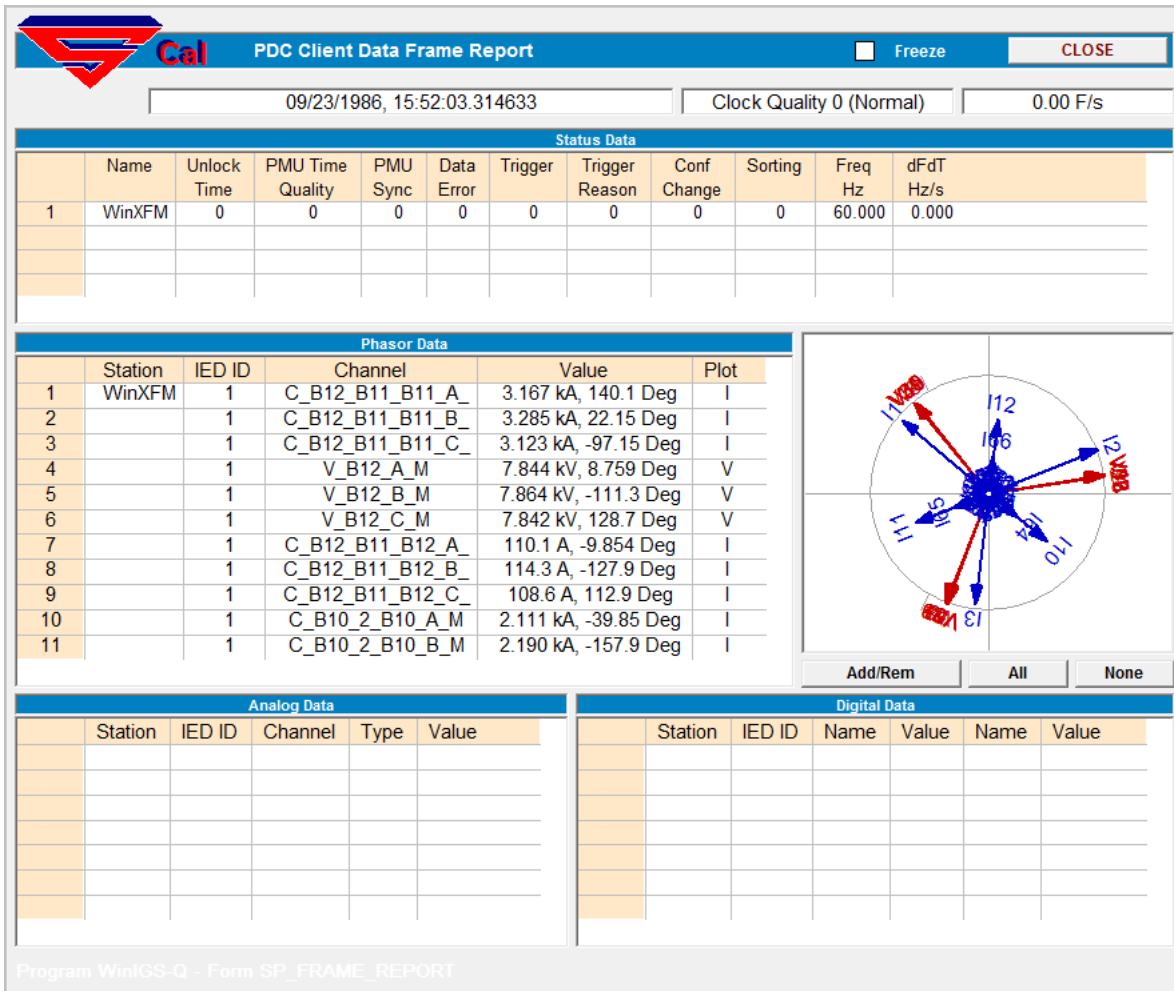


Figure 8.7: PDC Client Data Frame Report

Before the execution of the state estimator, the user has to map the measurements to the corresponding state estimator measurement channels in the PDC client as shown in Figure 8.8. This is a necessary step because the state estimator measurement channels are linked to the model of the system and have all the information that is necessary for the execution of the state estimator. And then, the state estimator identifies the topology, creates the derived measurements, pseudo-measurements, and virtual measurements automatically, and performs the quasi-dynamic state estimation. The estimated measurements, estimated states, residuals between measurements and estimated measurements, normalized residuals, variance of each state, chi-square, confidence level, and execution time can be displayed and output.

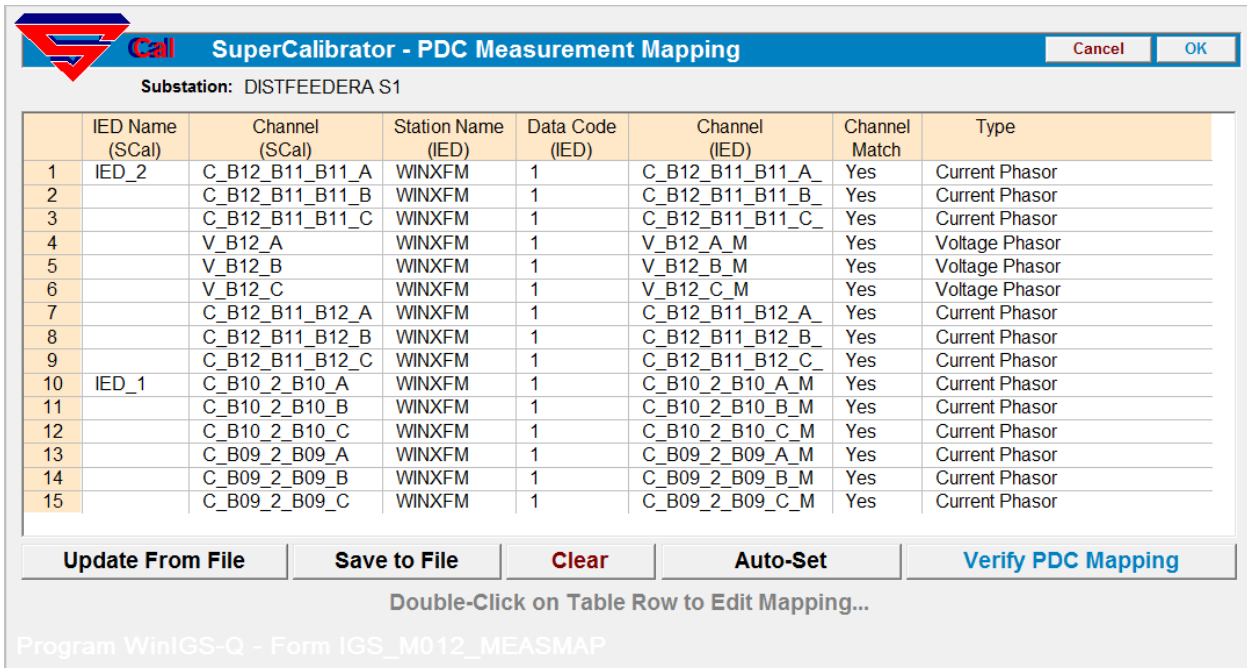


Figure 8.8: Measurement Mapping

Section 8.3: Test Case Results

The time plots of all the measurements from four relays are shown in Section 8.3. The estimated state report, measurement, and estimated measurement report are generated when the state estimator is running, the reports at one time stamp are shown in Figure 8.9, Figure 8.10 and Figure 8.11, respectively.

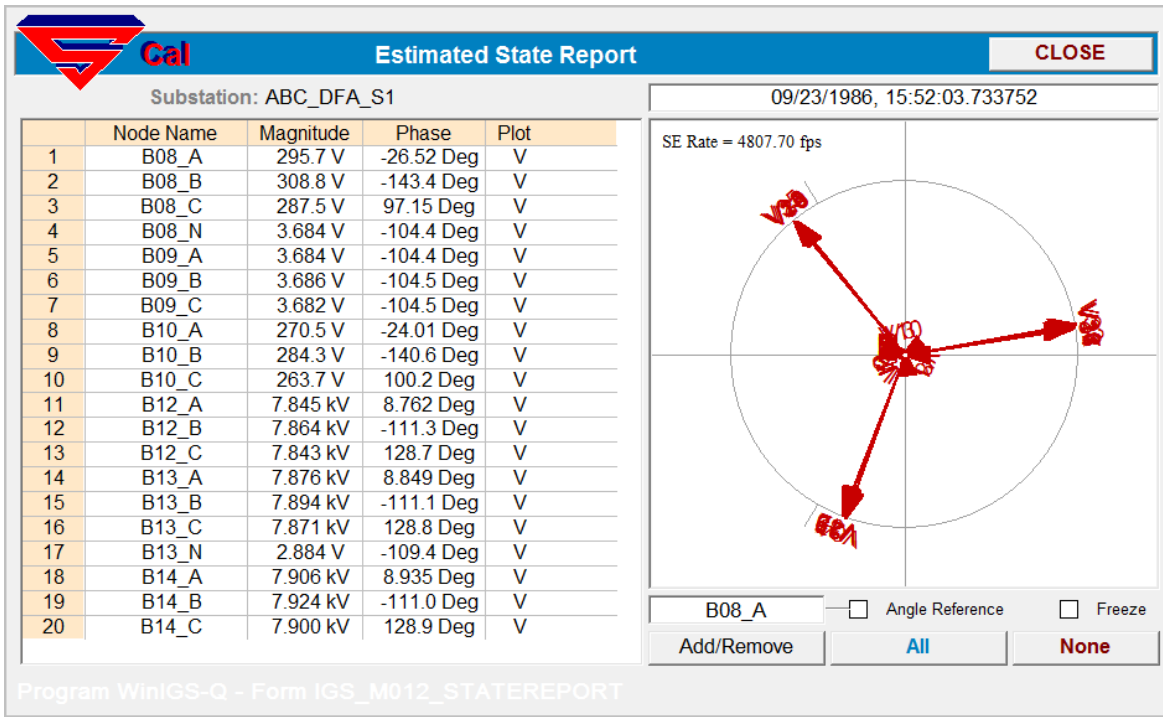


Figure 8.9: Estimated State Report

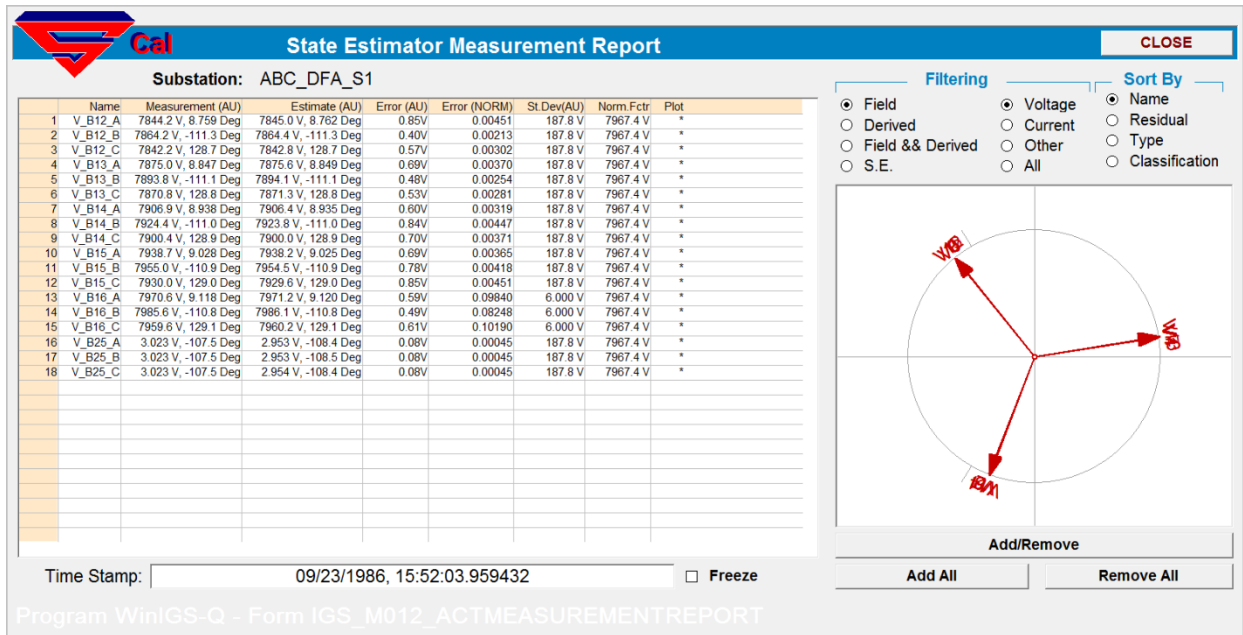


Figure 8.10: Estimated Voltage Measurement Report

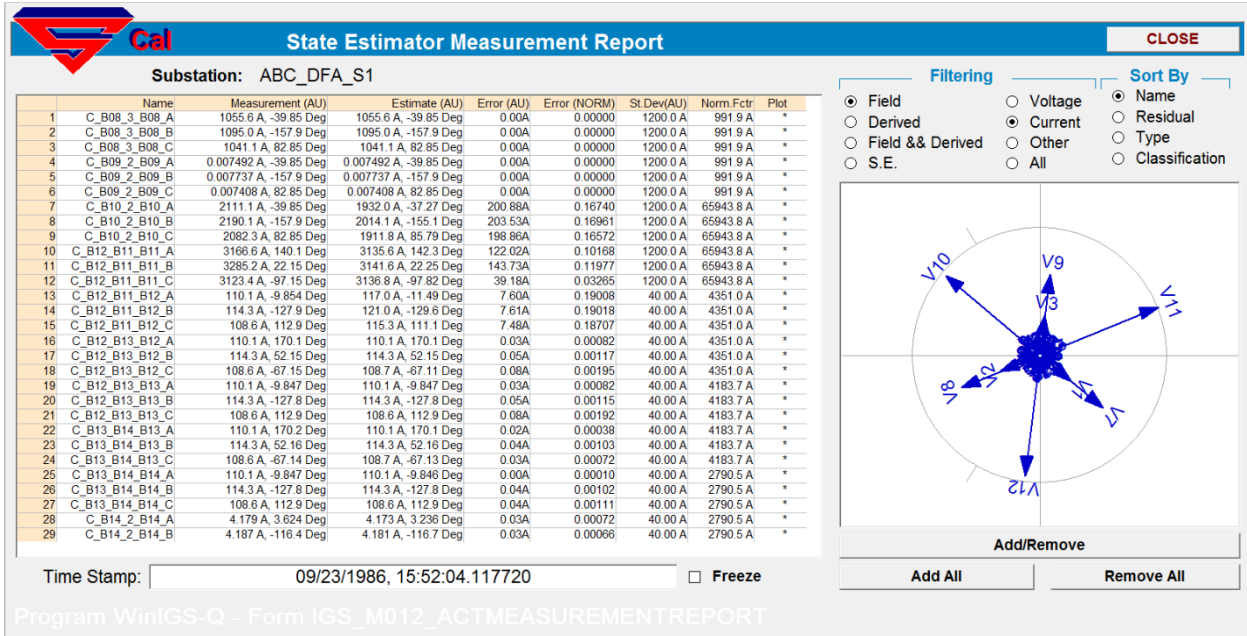


Figure 8.11: Estimated Current Measurements Report

The performance evaluation, which is also called the parameterized (parameter k) chi-square test is shown in Figure 8.12 and it is generated as follows:

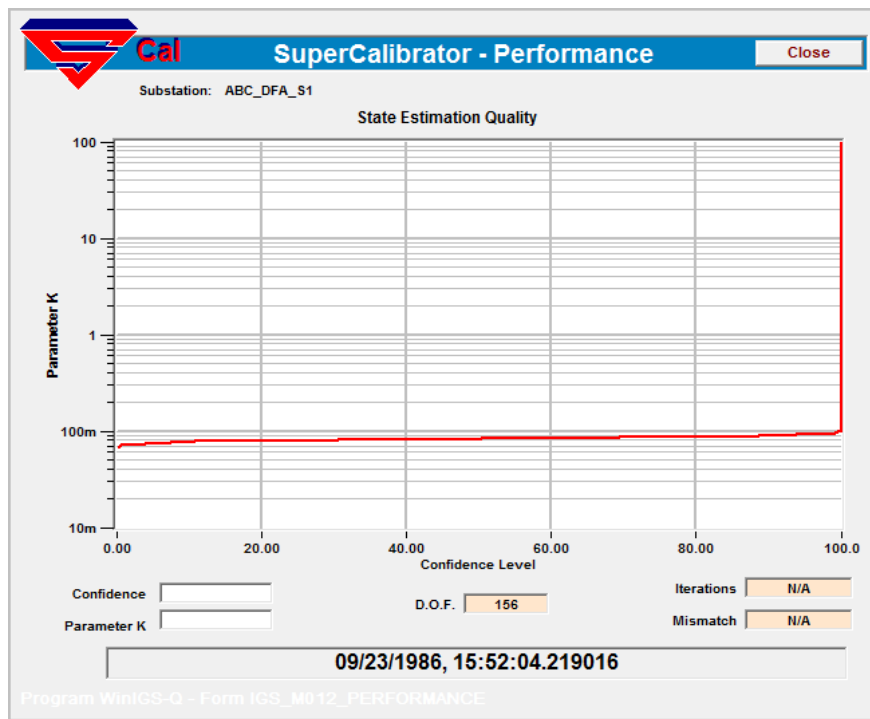


Figure 8.12: Performance Evaluation of State Estimation Result

All the phasors are divided into real and imaginary parts so that the state estimator is able to manipulate the data in real number. The chi-square test is calculated as:

$$\zeta = \sum_{i=1}^n \left(\frac{r_i}{k\sigma_i} \right)^2, \quad (8.14)$$

where r_i is the residual between measurement and estimated measurement i , and σ_i is the standard deviation of the corresponding measurement. Note that the variable k enables to express the results of the chi-square test with only one variable.

The confidence level is calculated as:

$$\Pr[\chi^2 \geq \zeta] = 1.0 - \Pr(\chi^2, \nu). \quad (8.15)$$

A confidence level around 100% (small chi-square value) when $k=1$ infers that the measurements are highly consistent with the dynamic model of the system, while a confidence level around 0% (large chi-square value) when $k=1$ means that the measurements do not match the dynamic model of the system. During this event, the confidence level is kept at 100% according to the parameterized chi-square test when $k=1$ as shown in Figure 8.12, which means that the measurements match the system model perfectly and there are no bad data or hidden failures in the system.

The state estimator also generates time plots of all the states. In this example, the states are the voltage phasors at each bus in this section, the internal states in some devices (such as the transformer), as well as the states at the other side of the interconnecting distribution line. The time plots of states at each bus are shown in Figure 8.13-8.18. Compared to the time plots of the actual voltage measurements at B12, B13, B16, and B25 as shown in Figure 8.19 and Figure 8.20, we find that the estimated states track the system perfectly.

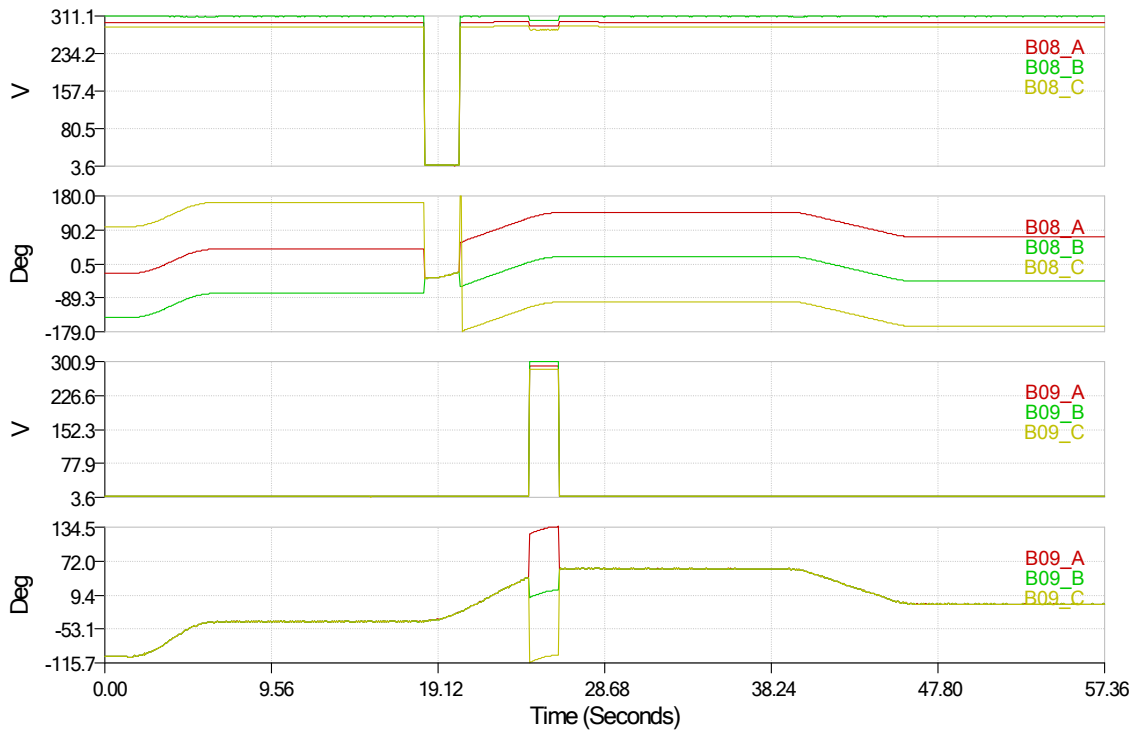


Figure 8.13: Estimated Voltage Phasors at B08 and B09

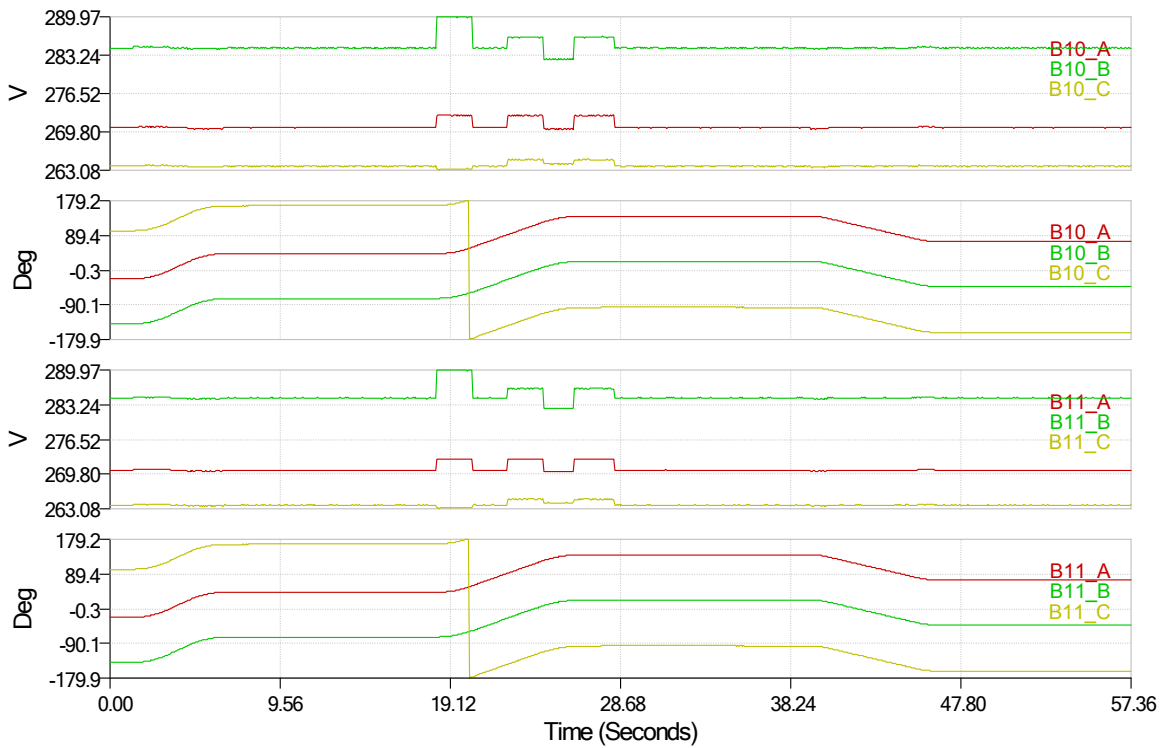


Figure 8.14: Estimated Voltage Phasors at B10 and B11

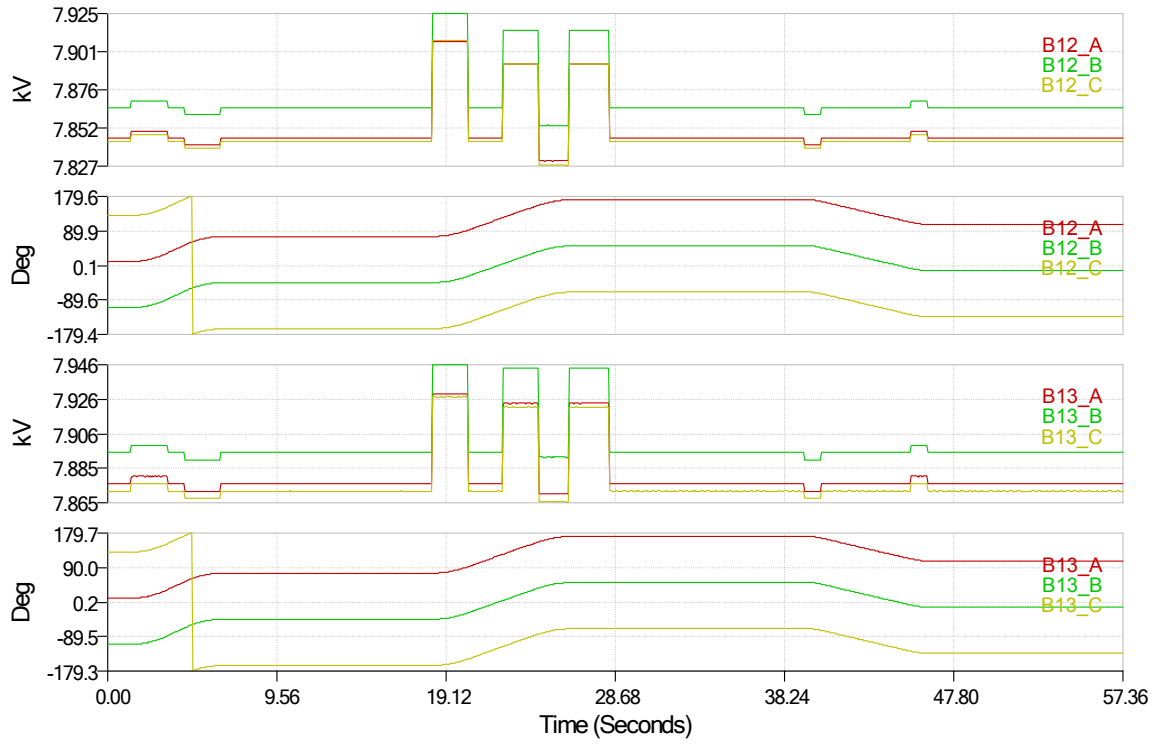


Figure 8.15: Estimated Voltage Phasors at B12 and B13

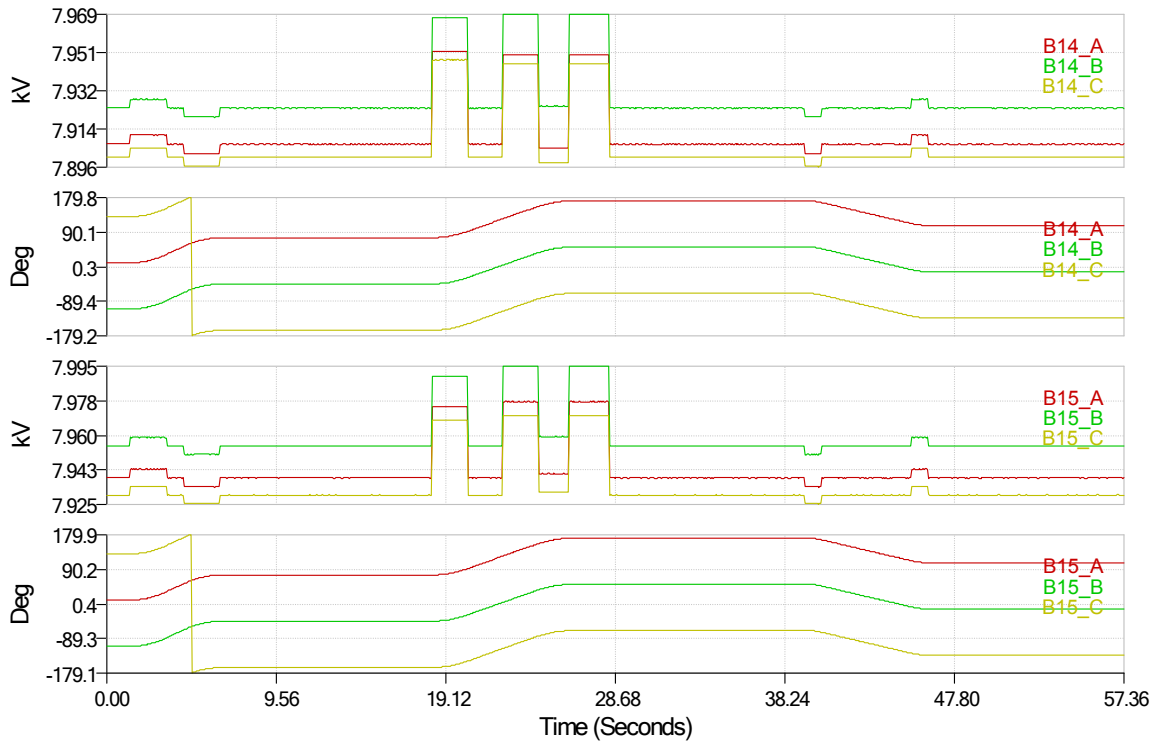


Figure 8.16: Estimated Voltage Phasors at B14 and B15

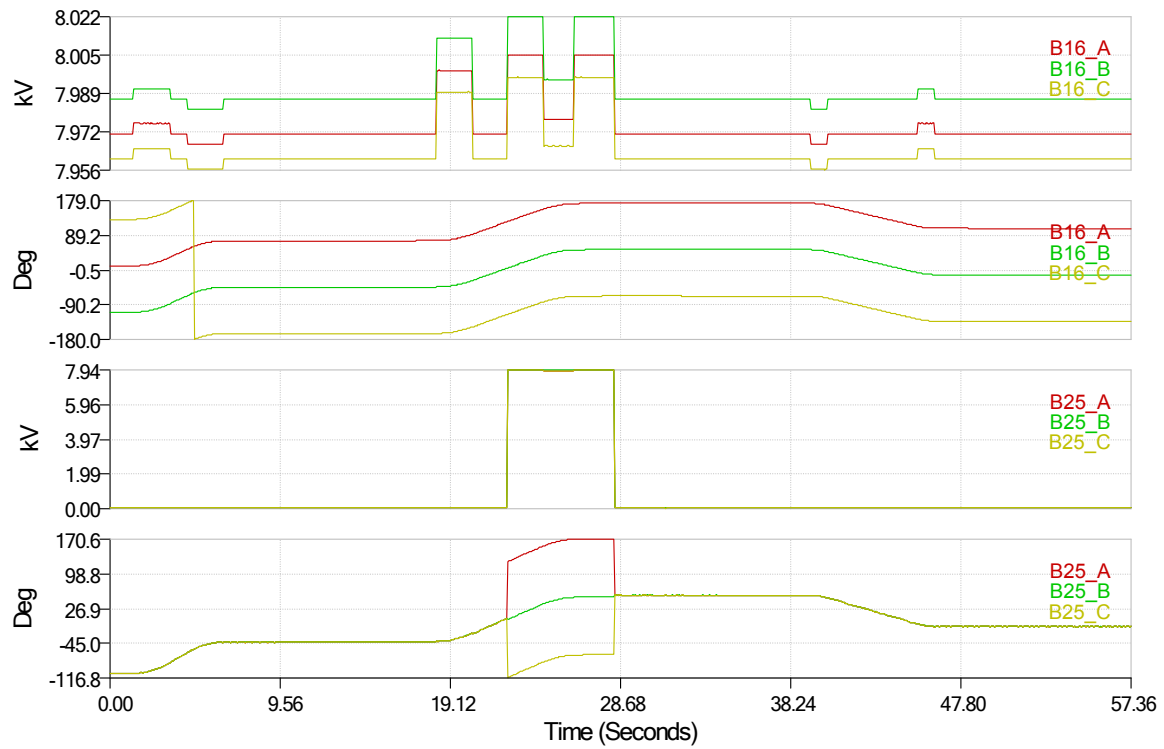


Figure 8.17: Estimated Voltage Phasors at B16 and B25

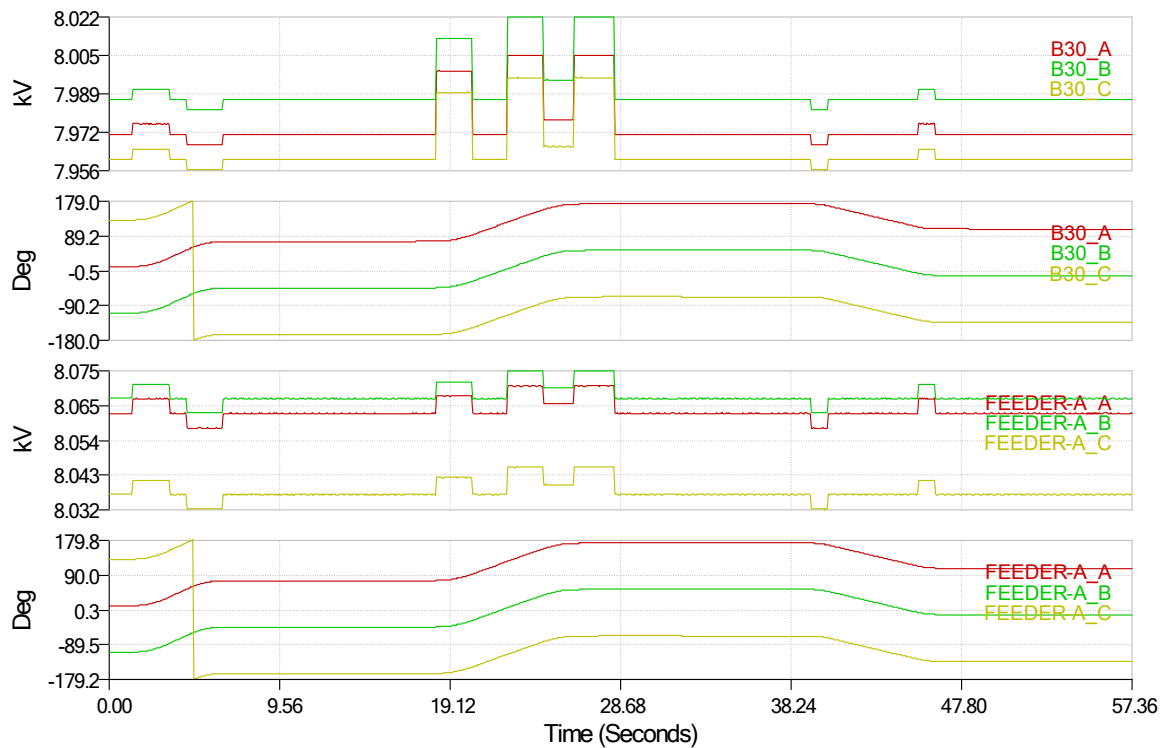


Figure 8.18: Estimated Voltage Phasors at B30 and Feeder-A

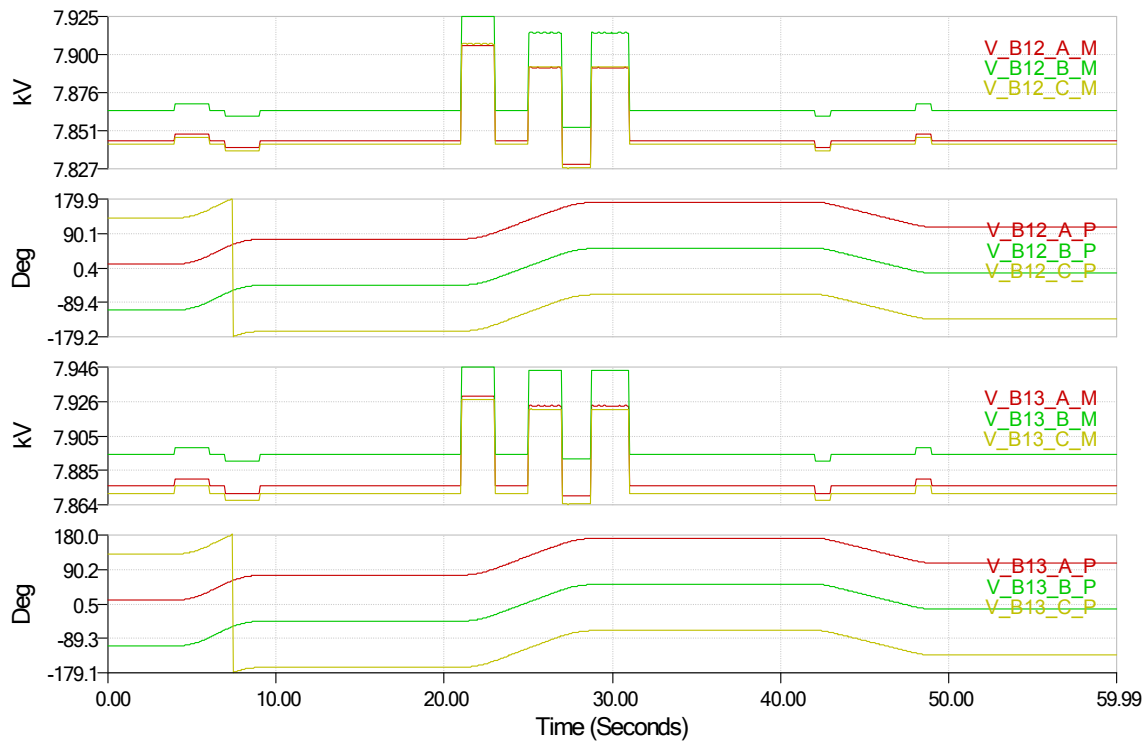


Figure 8.19: Voltage Measurements at B12 and B13

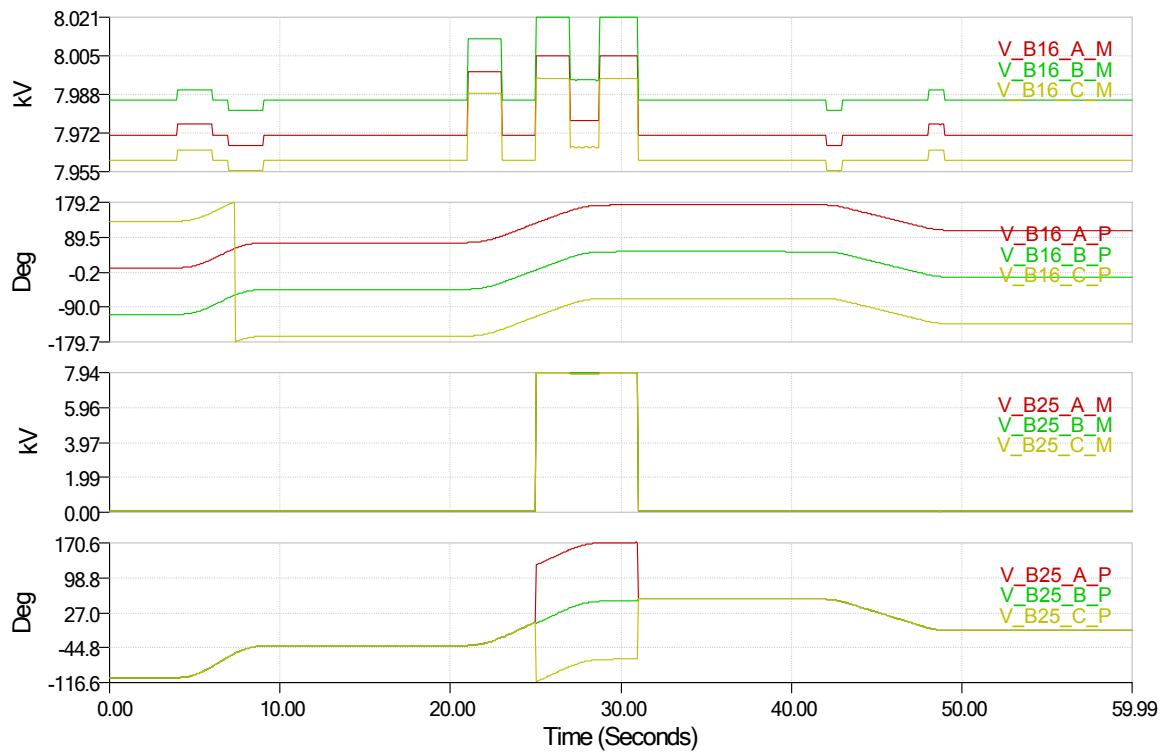


Figure 8.20: Voltage Measurements at B16 and B23

In addition, 3-D visualizations have been developed. A screenshot of the 3D visualization is shown in Figure 8.21. The estimated voltage magnitude for each node is visualized as a tube. The height of the tube is proportional to the voltage magnitude. The estimated voltage phase of a node is visualized as an arc. The angle of the arc is proportional to the voltage phase angle. Surface plots are also available as illustrated in Figure 8.21, the voltage measurements or estimated measurements are plotted as a contour map to reflect the magnitude of each measurement. Note that in this test case, the voltage surface at the load sides are a bit lower than the rated voltage because of the loads in this section.

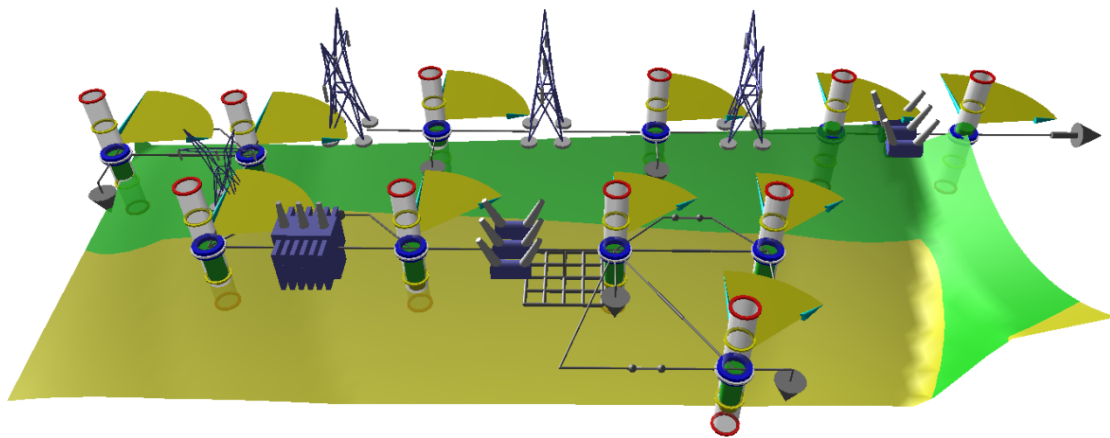


Figure 8.21: 3D Visualization Screenshot

Section 9: Summary and Conclusions

This report presents an object-oriented implementation of full state feedback control for VPPs. The components of the VPP full state feedback control are: (1) object-oriented high-fidelity modeling for all devices in the VPP; (2) DS-DQSE that continuously monitors and outputs estimated states and validated models by performing quasi-dynamic state estimation; (3) OPF solver that uses the output (estimated states and validated models) of DS-DQSE to formulate and solve the OPF which provides the optimal control commands as a feedback to the VPP.

The object-oriented device modeling approach has been presented in Section 3. The modeling approach starts from the physically based model with state and control variables. The end result is an algebraic model, referred to as state and control algebraic quadratic companion form (SCAQCF). The advantages of SCAQCF are (1) it is an object-oriented, interoperable, and unified syntax for all the devices in the power system; (2) It is easy to be formed and it stores all the information of a device; (3) DS-DQSE and OPF solver are able to work directly on these SCAQCF models without any other model information.

DS-DQSE, a critical role in full state feedback control system, is then proposed. DS-DQSE is able to provide real-time estimated states, validated measurements and validated models after quasi-dynamic state estimation. Besides, in addition to the actual data collected from IEDs, several other types of measurements (derived, pseudo, and virtual measurements) are defined, resulting in high measurement redundancy, which guarantees the accuracy of the estimated states and the network model of VPP. DS-DQSE is installed in a distributed architecture, which has the following advantages: (1) Since DS-DQSE uses local measurements to estimate the states in the local distribution system, the data traffic is confined; (2) The dimension of the problem solved by DS-DQSE is much smaller than that solved by the traditional centralized state estimator, and therefore, the execution time of the DS-DQSE is fast (i.e. at each cycle); (3) the relatively small dimension of the system allows very detailed power system models that can eliminate the estimation errors from the imbalanced operations and the asymmetric system, as well as the measurement errors introduced by the instrumentation channels.

An OPF solver is illustrated to solve the optimal control problem for VPPs. The OPF solver processes the output (estimated states and validated models) of DS-DQSE and gives the optimal control command as a feedback to the VPP. The OPF solver is also implemented in an object-oriented way and the algorithm of OPF solver proves to be robust and efficient. The robustness means that the algorithm starts from a feasible but not optimal solution and moves the operating point in the feasible region while approaching the optimality. Therefore, the output of the algorithm is always a feasible solution. High efficiency means that the algorithm consumes less runtime compared with traditional solution methods for the OPF problem. The reasons are as follows. Firstly, the algorithm models the OPF problem as a quadratic problem for fast convergence. Secondly, the algorithm identifies the active constraints gradually and adds them to the modeled constraint set if needed. These features ensure that at each iteration, the dimension of the problem is the smallest possible for the specific distribution system section.

Appendix A: Object-Oriented Modeling Example (Converter with P-Q Control)

This appendix provides the derivation of the IGBT converter average model with P-Q controller from compact device model to the SCAQCF model. The diagram of the P-Q control converter and the circuit diagram are given in Figure 3.2 and Figure 3.3. The states, controls, and parameters are introduced in Section 3.3. Based on these information, we first create the compact device model, and then generate the quadratized device model by introducing additional states to make the highest order less or equal to two. In the end, we present the SCAQCF model.

A.1: Converter Quasi-Dynamic Domain Compact Device Model

The DC-AC converter described here is an IGBT converter with pulse width modulation. Figure 3.3 shows the circuit diagram of the DC-AC converter with five terminals. $v_{AD}^{\%}$, $v_{KD}^{\%}$, $i_{AD}^{\%}$, and $i_{KD}^{\%}$ are voltages and currents at DC terminals while $v_a^{\%}$, $v_b^{\%}$, $v_c^{\%}$, $i_a^{\%}$, $i_b^{\%}$, and $i_c^{\%}$ are the voltages and currents at AC terminals. In addition, r is the resistance at DC side, L_s is the inductance at each phase on AC side, and $E_{DC}^{\%}$ is the voltage across the capacitor.

The through variables in this model are the five inflow terminal currents $i_{AD}^{\%}$, $i_{KD}^{\%}$, $i_a^{\%}$, $i_b^{\%}$, and $i_c^{\%}$.

The states are: $v_{AD}^{\%}$, $v_{KD}^{\%}$, $v_a^{\%}$, $v_b^{\%}$, $v_c^{\%}$, $E_{DC}^{\%}$, $E_a^{\%}$, $E_b^{\%}$, $E_c^{\%}$, P_{ac} , Q_{ac} , m , α .

The control variables are: P_{ref} , Q_{ref} .

The compact device model is therefore given by

$$i_{AD}^{\%} = \frac{v_{AD}^{\%} - v_{KD}^{\%} - E_{DC}^{\%}}{2r} \quad (\text{A.1})$$

$$i_{KD}^{\%} = \frac{-v_{AD}^{\%} + v_{KD}^{\%} + E_{DC}^{\%}}{2r} \quad (\text{A.2})$$

$$i_a^{\%} = \frac{1}{j\omega L_s} (v_a^{\%} - E_a^{\%}) \quad (\text{A.3})$$

$$i_b^{\%} = \frac{1}{j\omega L_s} (v_b^{\%} - E_b^{\%}) \quad (\text{A.4})$$

$$V_c^{\%} = \frac{1}{j\omega L_s} (V_c^{\%} - E_c^{\%}) \quad (\text{A.5})$$

$$0 = E_{DCr} I_{ADr} - P_{ac} \quad (\text{A.6})$$

$$0 = E_a^{\%} e^{j(2\pi/3)} - E_b^{\%} \quad (\text{A.7})$$

$$0 = E_a^{\%} e^{j(2\pi/3)} - E_c^{\%} \quad (\text{A.8})$$

$$0 = \text{Re} (V_a^{\%} I_a^{\%} + V_b^{\%} I_b^{\%} + V_c^{\%} I_c^{\%}) + P_{ac} \quad (\text{A.9})$$

$$0 = \text{Im} (V_a^{\%} I_a^{\%} + V_b^{\%} I_b^{\%} + V_c^{\%} I_c^{\%}) + Q_{ac} \quad (\text{A.10})$$

$$0 = K_{P1} \frac{d(P_{ref} - P_{ac})}{dt} + K_{I1} (P_{ref} - P_{ac}) - \frac{d \sin \alpha}{dt} \quad (\text{A.11})$$

$$0 = K_{P2} \frac{d(Q_{ref} - Q_{ac})}{dt} + K_{I2} (Q_{ref} - Q_{ac}) - \frac{dm}{dt} \quad (\text{A.12})$$

$$0 = \frac{m}{2\sqrt{2}} E_{DCr} \frac{V_a^{\%}}{|V_a^{\%}|} - E_a^{\%} e^{-j\alpha} \quad (\text{A.13})$$

where P_{ac} and Q_{ac} are the output active and reactive power of the converter. P_{ref} and Q_{ref} are the targeted active and reactive power output of the converter, m is the modulation index of the converter, α is the phase angle difference between $E_a^{\%}$ and $V_a^{\%}$, K_{P1} and K_{I1} are the proportional and integral coefficient of PQ controller for real power, K_{P2} and K_{I2} are the proportional and integral coefficient of PQ controller for reactive power. Note that phasors are used in the compact device model.

Operation Constraints:

$$0 \leq \frac{1}{2r} V_{AD}^{\%} - \frac{1}{2r} V_{KD}^{\%} - \frac{1}{2r} E_{DC}^{\%} \leq I_{DC,max} \quad (\text{A.c1})$$

$$0 \leq \left| \frac{1}{j\omega L_s} V_a^{\%} - \frac{1}{j\omega L_s} E_a^{\%} \right| \leq I_{AC,max} \quad (\text{A.c2})$$

$$0 \leq \left| \frac{1}{j\omega L_s} V_b^{\%} - \frac{1}{j\omega L_s} E_b^{\%} \right| \leq I_{AC,max} \quad (\text{A.c3})$$

$$0 \leq \left| \frac{1}{j\omega L_s} V_c - \frac{1}{j\omega L_s} E_c \right| \leq I_{AC,max} \quad (\text{A.c4})$$

$$0.0 \leq m \leq 1.0 \quad (\text{A.c5})$$

A.2: Converter Quasi-Dynamic Domain Quadratized Device Model

The compact device model is expanded into the quadratized device model (QDM). Additional states are added to guarantee the highest order in QDM is two. The QDM has 26 states and 26 equations, including 10 through equations and 16 internal equations.

The states are: V_{ADr} , V_{ADi} , V_{KDr} , V_{KDi} , V_{ar} , V_{ai} , V_{br} , V_{bi} , V_{cr} , V_{ci} , E_{DCr} , E_{DCi} , E_{ar} , E_{ai} , E_{br} , E_{bi} , E_{cr} , E_{ci} , P_{ac} , Q_{ac} , m , $V_{a,mag}$, mE_{DC} , $mE_{DC}OverV$, s_1 , s_2 .

The control variables are: P_{ref} , Q_{ref} .

The equations are listed as follows:

Equation Set 1 (linear through equations):

$$I_{ADr} = \frac{V_{ADr} - V_{KDr} - E_{DCr}}{2r} \quad (\text{A.14})$$

$$I_{ADi} = \frac{V_{ADi} - V_{KDi} - E_{DCi}}{2r} \quad (\text{A.15})$$

$$I_{KDr} = \frac{-V_{ADr} + V_{KDr} + E_{DCr}}{2r} \quad (\text{A.16})$$

$$I_{KDi} = \frac{-V_{ADi} + V_{KDi} + E_{DCi}}{2r} \quad (\text{A.17})$$

$$I_{ar} = \frac{1}{\omega L_s} (V_{ai} - E_{ai}) \quad (\text{A.18})$$

$$I_{ai} = -\frac{1}{\omega L_s} (V_{ar} - E_{ar}) \quad (\text{A.19})$$

$$I_{br} = \frac{1}{\omega L_s} (V_{bi} - E_{bi}) \quad (\text{A.20})$$

$$I_{bi} = -\frac{1}{\omega L_s} (V_{br} - E_{br}) \quad (\text{A.21})$$

$$I_{cr} = \frac{1}{\omega L_s} (V_{ci} - E_{ci}) \quad (\text{A.22})$$

$$I_{ci} = -\frac{1}{\omega L_s} (V_{cr} - E_{cr}) \quad (\text{A.23})$$

Equation Set 2 (linear internal equations):

$$0 = -\frac{1}{2} E_{ar} + \frac{\sqrt{3}}{2} E_{ai} - E_{br} \quad (\text{A.24})$$

$$0 = -\frac{\sqrt{3}}{2} E_{ar} - \frac{1}{2} E_{ai} - E_{bi} \quad (\text{A.25})$$

$$0 = -\frac{1}{2} E_{ar} - \frac{\sqrt{3}}{2} E_{ai} - E_{cr} \quad (\text{A.26})$$

$$0 = \frac{\sqrt{3}}{2} E_{ar} - \frac{1}{2} E_{ai} - E_{ci} \quad (\text{A.27})$$

$$0 = -K_{P1} \frac{dP_{ac}}{dt} + K_{I1} (P_{ref} - P_{ac}) - \frac{ds_1}{dt} \quad (\text{A.28})$$

$$0 = -K_{P2} \frac{dQ_{ac}}{dt} + K_{I2} (Q_{ref} - Q_{ac}) - \frac{dm}{dt} \quad (\text{A.29})$$

Equation Set 3 (quadratic equations):

$$0 = \frac{1}{2r} (V_{ADr} E_{DCr} - V_{KDr} E_{DCr} - E_{DCr}^2) - P_{ac} \quad (\text{A.30})$$

$$0 = E_{DCi} \quad (\text{A.31})$$

$$0 = \frac{1}{\omega L_s} (-V_{ar} E_{ai} + V_{ai} E_{ar} - V_{br} E_{bi} + V_{bi} E_{br} - V_{cr} E_{ci} + V_{ci} E_{cr}) + P_{ac} \quad (\text{A.32})$$

$$0 = \frac{1}{\omega L_s} (V_{ar}^2 - V_{ar} E_{ar} + V_{ai}^2 - V_{ai} E_{ai} + V_{br}^2 - V_{br} E_{br} + V_{bi}^2 - V_{bi} E_{bi} + V_{cr}^2 - V_{cr} E_{cr} + V_{ci}^2 - V_{ci} E_{ci}) + Q_{ac} \quad (\text{A.33})$$

$$0 = m \cdot E_{DC} - m E_{DC} \quad (\text{A.34})$$

$$0 = V_{ar}^2 + V_{ai}^2 - V_{a,mag}^2 \quad (\text{A.35})$$

$$0 = m E_{DC} - V_{a,mag} \cdot m E_{DC} \text{ Over } V \quad (\text{A.36})$$

$$0 = \frac{1}{2\sqrt{2}} m E_{DC} \text{Over} V \cdot V_{ar} - E_{ar} \cdot s_2 - E_{ai} \cdot s_1 \quad (\text{A.37})$$

$$0 = \frac{1}{2\sqrt{2}} m E_{DC} \text{Over} V \cdot V_{ai} - E_{ai} \cdot s_2 + E_{ar} \cdot s_1 \quad (\text{A.38})$$

$$0 = s_1^2 + s_2^2 - 1.0 \quad (\text{A.39})$$

Operation Constraints:

$$-I_{DC,\max} \leq \frac{1}{2r} V_{ADr} - \frac{1}{2r} V_{KDr} - \frac{1}{2r} E_{DCr} \leq I_{DC,\max} \quad (\text{A.c6})$$

$$0 \leq \frac{1}{\omega^2 L_s^2} V_{ar}^2 + \frac{1}{\omega^2 L_s^2} V_{ai}^2 + \frac{1}{\omega^2 L_s^2} E_{ar}^2 + \frac{1}{\omega^2 L_s^2} E_{ai}^2 - \frac{2}{\omega^2 L_s^2} V_{ar} E_{ar} - \frac{2}{\omega^2 L_s^2} V_{ai} E_{ai} \leq I_{AC,\max}^2 \quad (\text{A.c7})$$

$$0 \leq \frac{1}{\omega^2 L_s^2} V_{br}^2 + \frac{1}{\omega^2 L_s^2} V_{bi}^2 + \frac{1}{\omega^2 L_s^2} E_{br}^2 + \frac{1}{\omega^2 L_s^2} E_{bi}^2 - \frac{2}{\omega^2 L_s^2} V_{br} E_{br} - \frac{2}{\omega^2 L_s^2} V_{bi} E_{bi} \leq I_{AC,\max}^2 \quad (\text{A.c8})$$

$$0 \leq \frac{1}{\omega^2 L_s^2} V_{cr}^2 + \frac{1}{\omega^2 L_s^2} V_{ci}^2 + \frac{1}{\omega^2 L_s^2} E_{cr}^2 + \frac{1}{\omega^2 L_s^2} E_{ci}^2 - \frac{2}{\omega^2 L_s^2} V_{cr} E_{cr} - \frac{2}{\omega^2 L_s^2} V_{ci} E_{ci} \leq I_{AC,\max}^2 \quad (\text{A.c9})$$

$$0.0 \leq m \leq 1.0 \quad (\text{A.c10})$$

A.3: Converter Quasi-Dynamic Domain SCAQCF Device Model

The state and control algebraic quadratic form (SCAQCF) device model is derived from applying the quadratic integration to the quadratized device model with a time step h . The SCAQCF device model is:

$$\begin{Bmatrix} \mathbf{I}(t) \\ 0 \\ 0 \\ \mathbf{I}(t_m) \\ 0 \\ 0 \end{Bmatrix} = Y_{eqx} \mathbf{x} + Y_{equ} \mathbf{u} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{x}^T \langle F_{eqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T \langle F_{equ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \mathbf{u}^T \langle F_{equx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{Bmatrix} - B_{eq}$$

$$B_{eq} = -N_{eqx} \mathbf{x}(t-h) - N_{equ} \mathbf{u}(t-h) - M_{eq} \mathbf{I}(t-h) - K_{eq}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) = Y_{feqx} \mathbf{x} + Y_{fequ} \mathbf{u} + \left\{ \mathbf{x}^T \begin{matrix} \mathbf{M} \\ \langle F_{feqx}^i \rangle \\ \mathbf{M} \end{matrix} \mathbf{x} \right\} + \left\{ \mathbf{u}^T \begin{matrix} \mathbf{M} \\ \langle F_{fequ}^i \rangle \\ \mathbf{M} \end{matrix} \mathbf{u} \right\} + \left\{ \mathbf{u}^T \begin{matrix} \mathbf{M} \\ \langle F_{fequx}^i \rangle \\ \mathbf{M} \end{matrix} \mathbf{x} \right\} + C_{feq}$$

Connectivity: *TerminalNodeName*

$$\text{subject to: } \quad \mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{\max}$$

$$\quad \quad \quad \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$$

Normalization Factor: StateNormFactor, ThroughNormFactor, ControlNormFactor

Note: All the above variables are in metric system.

The normalization factors, functional constraints and variable limits are the same as the quasi-dynamic SCQDM.

where the matrices are given by

$$Y_{eqx} = \begin{bmatrix} \frac{4}{h} D_{eqxd1} + Y_{eqx1} & -\frac{8}{h} D_{eqxd1} \\ \frac{4}{h} D_{eqxd2} + Y_{eqx2} & -\frac{8}{h} D_{eqxd2} \\ Y_{eqx3} & 0 \\ \frac{1}{2h} D_{eqxd1} & \frac{2}{h} D_{eqxd1} + Y_{eqx1} \\ \frac{1}{2h} D_{eqxd2} & \frac{2}{h} D_{eqxd2} + Y_{eqx2} \\ 0 & Y_{eqx3} \end{bmatrix}$$

$$Y_{equ} = \begin{bmatrix} Y_{equ1} & 0 \\ Y_{equ2} & 0 \\ Y_{equ3} & 0 \\ 0 & Y_{equ1} \\ 0 & Y_{equ2} \\ 0 & Y_{equ3} \end{bmatrix} \quad F_{eqx} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ F_{eqxx3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & F_{eqxx3} \end{bmatrix} \quad F_{equ} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ F_{equu3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & F_{equu3} \end{bmatrix} \quad F_{equx} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ F_{equx3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & F_{equx3} \end{bmatrix}$$

$$N_{eqx} = \begin{bmatrix} -Y_{eqx1} + \frac{4}{h}D_{eqxd1} \\ -Y_{eqx2} + \frac{4}{h}D_{eqxd2} \\ 0 \\ \frac{1}{2}Y_{eqx1} - \frac{5}{2h}D_{eqxd1} \\ \frac{1}{2}Y_{eqx2} - \frac{5}{2h}D_{eqxd2} \\ 0 \end{bmatrix} \quad N_{equ} = \begin{bmatrix} -Y_{equ1} \\ -Y_{equ2} \\ 0 \\ \frac{1}{2}Y_{equ1} \\ \frac{1}{2}Y_{equ2} \\ 0 \end{bmatrix} \quad M_{eq} = \begin{bmatrix} I_{size(i(t))} \\ 0 \\ 0 \\ -\frac{1}{2}I_{size(i(t))} \\ 0 \\ 0 \end{bmatrix} \quad K_{eq} = \begin{bmatrix} 0 \\ 0 \\ C_{eqc3} \\ \frac{3}{2}C_{eqc1} \\ \frac{3}{2}C_{eqc2} \\ C_{eqc3} \end{bmatrix}$$

Appendix B: Object-Oriented Algorithm of Constructing the Network SCAQCF Model from Device SCAQCF Models

This appendix introduces the procedure to form the network SCAQCF model. The purpose of the network formation is to (1) provide the mapping lists (states, equations, controls) from devices to the network, (2) provide the network KCL equations at the common nodes. Given all the device models in this network and network interface node name list, the automatic construction of the network model is illustrated below.

Step 1: Input the device SCAQCF information and store the information into the variables defined in SCAQCF standard.

The SCAQCF standard appears in Section 3. This step has been done in device-level measurement model creation part.

Step 2: Input the network interface node name list.

Step 3: Creation of Mapping List from Device Nodes to Network Nodes. This step is achieved by

- (1) Creating a connector node name list;
- (2) Creating the network node model via breaker processing;
- (3) Creating a mapping list from device node to network node.

The result is stored in the array:

$pDev \rightarrow pQDSCAQCFModel_OptimalNodeNumber[i] = k;$

i: the node number in the device model

k: the node number in the network

Step 4: Determine the dimension of the network.

This step determines the dimension of all the arrays of the network model and then initializes these arrays.

Step 5: Mapping lists (equation, state, control, and constraint) creation.

The results are stored in the following arrays:

(1) Device Equation to Network Equation Mapping List

Example: $pDev = (Device^*)vImportedDevices[i]$
 $pDev \rightarrow pQDSCAQCFNetwork_EquIndex[j] = k;$
k: network equation number
i: device number
j: device equation number

(2) Device State to Network State Mapping List

Example: $pDev = (Device^*)vImportedDevices[i]$
 $pDev \rightarrow pQDSCAQCFNetwork_StateIndex[j] = k;$
k: network state number
i: device number
j: device state number

(3) Device Control Variable to Network Control Variable Mapping List

Example: $pDev = (Device^*)vImportedDevices[i]$
 $pDev \rightarrow pQDSCAQCFNetwork_ControlIndex[j] = k;$
k: network control number
i: device number
j: device control number

(4) Device Constraint to Network Constraint Mapping List

Example: $pDev = (Device^*)vImportedDevices[i]$
 $pDev \rightarrow pQDSCAQCFNetwork_ConstraintIndex[j] = k;$
k: network constraint number
i: device number
j: device constraint number

Step 6: Network model formation.

This step creates the network model by adding all the device contributions to the network.

(1) Create network state variable normalization factors.

Given:

$m_vdQDSCAQCFModel_StateNormFactor$ from $iDevice^{th}$ device
Device State to Network State Mapping List: $pQDSCAQCFNetwork_StateIndex$

Create:

$vdQDSCAQCFNetwork_StateNormFactor$

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in State Normalization Factor of the i_{dev} th device	Elements in State Normalization Factor of the Network
State number: i_{dev}	$i_{net} = \text{pDev} \rightarrow \text{pQDSCAQCFNetwork_StateIndex}[i_{dev}]$
Normalization factor: $v_{dev} = \text{pDev} \rightarrow \text{m_vdQDSCAQCFModel_StateNormFactor}[i_{dev}]$	$\text{vdQDSCAQCFNetwork_StateNormFactor}[i_{net}] =$ $\text{vdQDSCAQCFNetwork_StateNormFactor}[i_{net}] + v_{dev}$ $\text{vdQDSCAQCFNetwork_StateNormFactor}[i_{net}] =$ $\text{vdQDSCAQCFNetwork_StateNormFactor}[i_{net}] / n$

where n is the number of device states mapping to this network state.

(2) Formulate network through variable normalization factors

Given:

m_vdQDSCAQCFModel_ThroughNormFactor from i_{Device}^{th} device

Device Equation to Network Equation Mapping List: pQDSCAQCFNetwork_EquIndex

Create:

pQDSCAQCFNetwork_ThroughNormFactor

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Through Normalization Factor of the i_{Device}^{th} device	Elements in Through Normalization Factor of the Network
through variable number: i_{dev}	$i_{net} = \text{pDev} \rightarrow \text{pQDSCAQCFNetwork_EquIndex}[i_{dev}]$
normalization factor: $v_{dev} = \text{pDev} \rightarrow \text{m_vdQDSCAQCFModel_ThroughNormFactor}[i_{dev}]$	$\text{pQDSCAQCFNetwork_StateNormFactor}[i_{net}] =$ $\text{pQDSCAQCFNetwork_StateNormFactor}[i_{net}] + v_{dev}$ $\text{pQDSCAQCFNetwork_StateNormFactor}[i_{net}] =$

	$= \frac{\text{pQDSCAQCFNetwork_StateNormFactor}[i_{net}]}{n}$
--	---

where n is the number of device through variables mapping to this network through variables.

(3) Formulate network control variable normalization factors

Given:
 m_vdQDSCAQCFModel_ControlNormFactor from *iDevice*th device
 Device Control Variable to Network Control Variable Mapping List:
 pQDSCAQCFNetwork_ControlIndex

Create:
 vdQDSCAQCFNetwork_ControlNormFactor

Process:
 pDev=(Device*)vImportedDevices[iDevice];

Elements in Control Variable Normalization Factor of the <i>iDevice</i> th device	Elements in Control Variable Normalization Factor of the Network
control variable number: i_{dev}	$i_{net} = \text{pDev} \rightarrow$ vdQDSCAQCFNetwork_ControlIndex[i_{dev}]
normalization factor: $v_{dev} = \text{pDev} \rightarrow$ m_vdQDSCAQCFModel_ControlNormFactor[i_{dev}]	vdQDSCAQCFNetwork_ControlNormFactor[i_{net}] = v_{dev}

(4) Formulate network Yeqx

Given:
 m_vqQDSCAQCFModel_Yeqx from *iDevice*th device
 Device Equation to Network Equation Mapping List: pQDSCAQCFNetwork_EquIndex
 Device State to Network State Mapping List:
 pQDSCAQCFNetwork_StateIndex

Create:
 pQDSCAQCFNetwork_Yeqx

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Yeqx of the $iDevice^{th}$ device	Elements in Yeqx of the Network
equation number: i_{dev}	$i_{net} = pDev \rightarrow$ pQDSCAQCFNetwork_EquIndex[i_{dev}]
state number: j_{dev}	$j_{net} = pDev \rightarrow$ pQDSCAQCFNetwork_StateIndex[j_{dev}]
coefficient: $v_{dev} = pDev \rightarrow$ m_vqQDSCAQCFModel_Yeqx[i_{dev}][j_{dev}]	pQDSCAQCFNetwork_Yeqx[i_{net}][j_{net}] = pQDSCAQCFNetwork_Yeqx[i_{net}][j_{net}]+ v_{dev}

(5) Formulate network Yequ**Given:**

m_vqQDSCAQCFModel_Yequ from $iDevice^{th}$ device

Device Equation to Network Equation Mapping List: pQDSCAQCFNetwork_EquIndex

Device Control Variable to Network Control Variable Mapping List:

pQDSCAQCFNetwork_ControlIndex

Create:

pQDSCAQCFNetwork_Yequ

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Yequ of the $iDevice^{th}$ device	Elements in Yequ of the Network
equation number: i_{dev}	$i_{net} = pDev \rightarrow$ pQDSCAQCFNetwork_EquIndex[i_{dev}]
control number: j_{dev}	$j_{net} = pDev \rightarrow$ pQDSCAQCFNetwork_ControlIndex[j_{dev}]
coefficient: $v_{dev} = pDev \rightarrow$ m_vqQDSCAQCFModel_Yequ[i_{dev}][j_{dev}]	pQDSCAQCFNetwork_Yequ[i_{net}][j_{net}] = v_{dev}

(6) Formulate network Feqxx**Given:**

m_vcQDSCAQCFModel_Feqxx from $iDevice^{th}$ device

Device Equation to Network Equation Mapping List:

pQDSCAQCFNetwork_EquIndex

Device State Variable to Network State Variable Mapping List:

pQDSCAQCFNetwork_StateIndex

Create:

pQDSCAQCFNetwork_Feqxx

Process:

pDev=(Device*)vImportedDevices[iDevice]

Elements in Feqxx of the $iDevice^{th}$ device	Elements in Feqxx of the Network
equation number: k_{dev}	pQDSCAQCFNetwork_Feqxx[iFeqxx].scubix_k = pDev-> pQDSCAQCFNetwork_EquIndex[k_{dev}];
state number: i_{dev}	pQDSCAQCFNetwork_Feqxx[iFeqxx].scubix_i = pDev-> pQDSCAQCFNetwork_StateIndex[i_{dev}]
state number: j_{dev}	pQDSCAQCFNetwork_Feqxx[iFeqxx].scubix_j = pDev-> pQDSCAQCFNetwork_StateIndex[j_{dev}]
Coefficient: v_{dev}	pQDSCAQCFNetwork_Feqxx[iFeqxx].scubix_v= v_{dev} ; iFeqxx++;

(7) Formulate network Fequu

Given:

m_vcQDSCAQCFModel_Fequu from $iDevice^{th}$ device

Device Equation to Network Equation Mapping List:

pQDSCAQCFNetwork_EquIndex

Device Control Variable to Network Control Variable Mapping List:

pQDSCAQCFNetwork_ControlIndex

Create:

pQDSCAQCFNetwork_Fequu

Process:

pDev=(Device*)vImportedDevices[iDevice]

Elements in Fequu of the	Elements in Fequu of the Network
--------------------------	----------------------------------

$iDevice^{th}$ device	
equation number: k_{dev}	$pQDSCAQCFNetwork_Fequu[iFequu].scubix_k = pDev \rightarrow pQDSCAQCFNetwork_EquIndex[k_{dev}];$
control number: i_{dev}	$pQDSCAQCFNetwork_Fequu[iFequu].scubix_i = pDev \rightarrow pQDSCAQCFNetwork_ControlIndex[i_{dev}];$
control number: j_{dev}	$pQDSCAQCFNetwork_Fequu[iFequu].scubix_j = pDev \rightarrow pQDSCAQCFNetwork_ControlIndex[j_{dev}];$
Coefficient: v_{dev}	$pQDSCAQCFNetwork_Fequu[iFequu].scubix_v = v_{dev};$ $iFequu++;$

(8) Formulate network Fequx

Given:

$m_vcQDSCAQCFModel_Fequx$ from $iDevice^{th}$ device

Device Equation to Network Equation Mapping List: $pQDSCAQCFNetwork_EquIndex$

Device Control Variable to Network Control Variable Mapping List:

$pQDSCAQCFNetwork_ControlIndex$

Device State Variable to Network State Variable Mapping List:

$pQDSCAQCFNetwork_StateIndex$

Create:

$pQDSCAQCFNetwork_Fequx$

Process:

$pDev = (Device^*)vImportedDevices[iDevice]$

Elements in Fequx of the $iDevice^{th}$ device	Elements in Fequx of the Network
equation number: k_{dev}	$pQDSCAQCFNetwork_Fequx[iFequx].scubix_k = pDev \rightarrow pQDSCAQCFNetwork_EquIndex[k_{dev}];$
control number: i_{dev}	$pQDSCAQCFNetwork_Fequx[iFequx].scubix_i = pDev \rightarrow pQDSCAQCFNetwork_ControlIndex[i_{dev}];$
state number: j_{dev}	$pQDSCAQCFNetwork_Fequx[iFequx].scubix_j = pDev \rightarrow pQDSCAQCFNetwork_StateIndex[j_{dev}];$
coefficient: v_{dev}	$pQDSCAQCFNetwork_Fequx[iFequx].scubix_v = v_{dev};$

	iFequx++;
--	-----------

(9) Formulate network Neqx

Given:

m_vqQDSCAQCFModel_Neqx from *iDevice*th device
 Device Equation to Network Equation Mapping List: pQDSCAQCFNetwork_EquIndex
 Device State Variable to Network State Variable Mapping List:
 pQDSCAQCFNetwork_StateIndex

Create:

pQDSCAQCFNetwork_Neqx

Process:

pDev=(Device*)vImportedDevices[iDevice]

Elements in Neqx of the <i>iDevice</i> th device	Elements in Neqx of the Network
equation number: k_{dev}	$k_{net} = \text{pDev} \rightarrow$ pQDSCAQCFNetwork_EquIndex[k_{dev}]
state number: i_{dev}	$i_{net} = \text{pDev} \rightarrow$ pQDSCAQCFNetwork_StateIndex[i_{dev}]
coefficient: $v_{dev} = \text{pDev} \rightarrow$ m_vqQDSCAQCFModel_Neqx[k_{dev}][i_{dev}]	pQDSCAQCFNetwork_Neqx[k_{net}][i_{net}] = $v_{dev} +$ pQDSCAQCFNetwork_Neqx[k_{net}][i_{net}]

(10) Formulate network Nequ

Given:

m_vqQDSCAQCFModel_Nequ from *iDevice*th device
 Device Equation to Network Equation Mapping List: pQDSCAQCFNetwork_EquIndex
 Device Control Variable to Network Control Variable Mapping List:
 pQDSCAQCFNetwork_ControlIndex

Create:

pQDSCAQCFNetwork_Nequ

Process:

pDev=(Device*)vImportedDevices[iDevice]

Elements in Nequ of the <i>iDevice</i> th device	Elements in Nequ of the Network
---	---------------------------------

equation number: k_{dev}	$k_{net} = \text{pDev} \rightarrow$ pQDSCAQCFNetwork_EquIndex[k_{dev}]
control number: i_{dev}	$i_{net} = \text{pDev} \rightarrow$ pQDSCAQCFNetwork_ControlIndex[i_{dev}]
coefficient: $v_{dev} = \text{pDev} \rightarrow$ m_vqQDSCAQCFModel_Nequ[k_{dev}][i_{dev}]	pQDSCAQCFNetwork_Nequ[k_{net}][i_{net}] = v_{dev}

(11) Formulate network Meq

Given:

m_vqQDSCAQCFModel_Meq from $iDevice^{\text{th}}$ device
Device Equation to Network Equation Mapping List:
pQDSCAQCFNetwork_EquIndex

Create:

pQDSCAQCFNetwork_Meq

Process:

pDev=(Device*)vImportedDevices[iDevice]

Elements in Meq of the $iDevice^{\text{th}}$ device	Elements in Meq of the Network
equation number: k_{dev}	$k_{net} = \text{pDev} \rightarrow$ pQDSCAQCFNetwork_EquIndex[k_{dev}]
equation number: i_{dev}	$i_{net} = \text{pDev} \rightarrow$ pQDSCAQCFNetwork_EquIndex[i_{dev}]
coefficient: $v_{dev} = \text{pDev} \rightarrow$ m_vqQDSCAQCFModel_Meq[k_{dev}][i_{dev}]	pQDSCAQCFNetwork_Meq[k_{net}][i_{net}] = v_{dev} + pQDSCAQCFNetwork_Meq[k_{net}][i_{net}]

(12) Formulate network Keq

Given:

m_vlQDSCAQCFModel_Keq from $iDevice^{\text{th}}$ device
Device Equation to Network Equation Mapping List:
pQDSCAQCFNetwork_EquIndex

Create:

pQDSCAQCFNetwork_Keq

Process:

pDev=(Device*)vImportedDevices[iDevice]

Elements in Keq of the $iDevice^{th}$ device	Elements in Keq of the Network
equation number: k_{dev}	$k_{net} = pDev \rightarrow pQDSCAQCFNetwork_EquIndex[k_{dev}]$
coefficient: $v_{dev} = pDev \rightarrow m_vIQDSCAQCFModel_Keq[k_{dev}]$	$pQDSCAQCFNetwork_Keq[k_{net}] = v_{dev} + pQDSCAQCFNetwork_Keq[k_{net}]$

(13) Formulate network Yfeqx

Given:

$m_vIQDSCAQCFModel_Yfeqx$ from $iDevice^{th}$ device

Device Constraint Equation to Network Constraint Equation Mapping List:

$pQDSCAQCFNetwork_ConstraintIndex$

Device State to Network State Mapping List:

$pQDSCAQCFNetwork_StateIndex$

Create:

$pQDSCAQCFNetwork_Yfeqx$

Process:

$pDev = (Device*)vImportedDevices[iDevice];$

Elements in Yfeqx of the $iDevice^{th}$ device	Elements in Yfeqx of the Network
constraint equation number: i_{dev}	$i_{net} = pDev \rightarrow pQDSCAQCFNetwork_ConstraintIndex[i_{dev}]$
state number: j_{dev}	$j_{net} = pDev \rightarrow pQDSCAQCFNetwork_StateIndex[j_{dev}]$
coefficient: $v_{dev} = pDev \rightarrow m_vIQDSCAQCFModel_Yfeqx[i_{dev}][j_{dev}]$	$pQDSCAQCFNetwork_Yfeqx[i_{net}][j_{net}] = v_{dev};$

(14) Formulate network Yfequ

Given:

$m_vIQDSCAQCFModel_Yfequ$ from $iDevice^{th}$ device

Device Constraint Equation to Network Constraint Equation Mapping List:

$pQDSCAQCFNetwork_ConstraintIndex$

Device Control to Network Control Mapping List:

$pQDSCAQCFNetwork_ControlIndex$

Create:

pQDSCAQCFNetwork_Yfequ

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Yfequ of the $iDevice^{th}$ device	Elements in Yfequ of the Network
constraint equation number: i_{dev}	$i_{net} = pDev \rightarrow$ pQDSCAQCFNetwork_ConstraintIndex[i_{dev}]
control number: j_{dev}	$j_{net} = pDev \rightarrow$ pQDSCAQCFNetwork_ControlIndex[j_{dev}]
coefficient: $v_{dev} = pDev \rightarrow$ >m_vqQDSCAQCFModel_Yfequ[i_{dev}][j_{dev}]	pQDSCAQCFNetwork_Yfequ[i_{net}][j_{net}] = v_{dev} ;

(15) Formulate network Ffeqxx**Given:**

m_vcQDSCAQCFModel_Ffeqxx from $iDevice^{th}$ device

Device Constraint Equation to Network Constraint Equation Mapping List:

pQDSCAQCFNetwork_ConstraintIndex

Device State Variable to Network State Variable Mapping List:

pQDSCAQCFNetwork_StateIndex

Create:

pQDSCAQCFNetwork_Ffeqxx

Process:

pDev=(Device*)vImportedDevices[iDevice]

Elements in Ffeqxx of the $iDevice^{th}$ device	Elements in Ffeqxx of the Network
constraint equation number: k_{dev}	pQDSCAQCFNetwork_Ffeqxx[iFfeqxx].scubix_k = pDev-> pQDSCAQCFNetwork_ConstraintIndex[k_{dev}];
state number: i_{dev}	pQDSCAQCFNetwork_Ffeqxx[iFfeqxx].scubix_i = pDev-> pQDSCAQCFNetwork_StateIndex[i_{dev}]
state number: j_{dev}	pQDSCAQCFNetwork_Ffeqxx[iFfeqxx].scubix_j = pDev-> pQDSCAQCFNetwork_StateIndex[j_{dev}]

Coefficient: v_{dev}	<p> <code>pQDSCAQCFNetwork_Ffeqxx[iFfeqxx].scubix_v= v_{dev};</code> <code>iFfeqxx++;</code> </p>
------------------------	---

(16) Formulate network Ffequu

Given:

m_{vc} QDSCAQCFModel_Ffequu from $iDevice^{th}$ device
 Device Constraint Equation to Network Constraint Equation Mapping List:
 pQDSCAQCFNetwork_ConstraintIndex
 Device Control Variable to Network Control Variable Mapping List:
 pQDSCAQCFNetwork_ControlIndex

Create:

pQDSCAQCFNetwork_Ffequu

Process:

pDev=(Device*)vImportedDevices[iDevice]

Elements in Ffequu of the $iDevice^{th}$ device	Elements in Ffequu of the Network
constraint equation number: k_{dev}	<code>pQDSCAQCFNetwork_Ffequu[iFfequu].scubix_k =</code> <code>pDev-> pQDSCAQCFNetwork_ConstraintIndex[k_{dev}];</code>
control number: i_{dev}	<code>pQDSCAQCFNetwork_Ffequu[iFfequu].scubix_i =</code> <code>pDev-> pQDSCAQCFNetwork_ControlIndex[i_{dev}]</code>
control number: j_{dev}	<code>pQDSCAQCFNetwork_Ffequu[iFfequu].scubix_j =</code> <code>pDev-> pQDSCAQCFNetwork_ControlIndex[j_{dev}]</code>
Coefficient: v_{dev}	<code>pQDSCAQCFNetwork_Ffequu[iFfequu].scubix_v= v_{dev};</code> <code>iFfequu++;</code>

(17) Formulate network Ffequx

Given:

m_{vc} QDSCAQCFModel_Ffequx from $iDevice^{th}$ device
 Device Constraint Equation to Network Constraint Equation Mapping List:
 pQDSCAQCFNetwork_ConstraintIndex
 Device Control Variable to Network Control Variable Mapping List:
 pQDSCAQCFNetwork_ControlIndex

Device State Variable to Network State Variable Mapping List:
 pQDSCAQCFNetwork_StateIndex

Create:

pQDSCAQCFNetwork_Ffequx

Process:

pDev=(Device*)vImportedDevices[iDevice]

Elements in Ffequx of the $iDevice^{th}$ device	Elements in Ffequx of the Network
constraint equation number: k_{dev}	pQDSCAQCFNetwork_Ffequx[iFfequx].scubix_k = pDev-> pQDSCAQCFNetwork_ConstraintIndex[k_{dev}];
control number: i_{dev}	pQDSCAQCFNetwork_Ffequx[iFfequx].scubix_i = pDev-> pQDSCAQCFNetwork_ControlIndex[i_{dev}]
state number: j_{dev}	pQDSCAQCFNetwork_Ffequx[iFfequx].scubix_j = pDev-> pQDSCAQCFNetwork_StateIndex[j_{dev}]
Coefficient: v_{dev}	pQDSCAQCFNetwork_Ffequx[iFfequx].scubix_v= v_{dev} ; iFfequx++;

(18) Formulate network Cfeq

Given:

m_vlQDSCAQCFModel_Cfeq from $iDevice^{th}$ device

Device Constraint Equation to Network Constraint Equation Mapping List:

pQDSCAQCFNetwork_ConstraintIndex

Create:

pQDSCAQCFNetwork_Cfeq

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Cfeq of the $iDevice^{th}$ device	Elements in Cfeq of the Network
constraint equation number: i_{dev}	$i_{net} =$ pDev-> pQDSCAQCFNetwork_ConstraintIndex[i_{dev}]
coefficient: $v_{dev} =$ pDev-> m_vlQDSCAQCFModel_Cfeq[i_{dev}]	pQDSCAQCFNetwork_Cfeq[i_{net}] = v_{dev} ;

The general expression of the network SCAQCF model is:

$$\begin{Bmatrix} \mathbf{I}(t) \\ 0 \\ 0 \\ \mathbf{I}(t_m) \\ 0 \\ 0 \end{Bmatrix} = Y_{net,eqx} \mathbf{x} + Y_{net,equ} \mathbf{u} + \begin{Bmatrix} \mathbf{M} \\ \langle F_{net,eqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \langle F_{net,equ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \langle F_{net,equx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{Bmatrix} - B_{net,eq}$$

$$B_{net,eq} = -N_{net,eqx} \mathbf{x}(t-h) - N_{net,equ} \mathbf{u}(t-h) - M_{net,eq} \mathbf{I}(t-h) - K_{net,eq}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) = Y_{net,feqx} \mathbf{x} + Y_{net,fequ} \mathbf{u} + \begin{Bmatrix} \mathbf{M} \\ \langle F_{net,feqx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \langle F_{net,fequ}^i \rangle \mathbf{u} \\ \mathbf{M} \end{Bmatrix} + \begin{Bmatrix} \mathbf{M} \\ \langle F_{net,fequx}^i \rangle \mathbf{x} \\ \mathbf{M} \end{Bmatrix} + C_{net,feq}$$

Connectivity: *TerminalNodeName*

$$\text{subject to: } \quad \mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{\max}$$

$$\quad \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$$

Normalization Factor: StateNormFactor, ThroughNormFactor, ControlNormFactor

Note: All the above variables are in metric system.

where

$\mathbf{I}(t)$ and $\mathbf{I}(t_m)$: the through variables of the network model;

\mathbf{x} : external and internal state variables of the network model, $\mathbf{x} = [\mathbf{x}(t), \mathbf{x}(t_m)]$;

\mathbf{u} : control variables of the network model, $\mathbf{u} = [\mathbf{u}(t), \mathbf{u}(t_m)]$;

$Y_{net,eqx}$: matrix defining the linear part for state variables;

$F_{net,eqx}$: matrices defining the quadratic part for state variables;

$Y_{net,equ}$: matrix defining the linear part for control variables;

$F_{net,equ}$: matrices defining the quadratic part for control variables;

$F_{net,equx}$: matrices defining the quadratic part for the product of state and control variables;

$B_{net,eq}$: history dependent vector of the network model;

$N_{net,eqx}$: matrix defining the last integration step state variables part;

$N_{net,equ}$: matrix defining the last integration step control variables part;

$M_{net,eq}$: matrix defining the last integration step through variables part;

$K_{net,eq}$: constant vector of the network model;

TerminalNodeName: terminal names defining the connectivity of the network model;

StateNormFactor: Normalization Factors for the states;

ThroughNormFactor: Normalization Factors for the through and zero variables;

ControlNormFactor: Normalization Factors for the controls;

$\mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{\max}$: operating constraints,

$\mathbf{u}_{\min}, \mathbf{u}_{\max}$: lower and upper bounds for the control variables;

$Y_{net,feqx}$: constraint matrix defining the linear part for state variables;

$F_{net,feqx}$: constraint matrices defining the quadratic part for state variables;

$Y_{net,fequ}$: constraint matrix defining the linear part for control variables;

$F_{net,fequ}$: constraint matrices defining the quadratic part for control variables;

$F_{net,fequx}$: constraint matrices defining the quadratic part for the product of state and control variables;

$C_{net,feq}$: constraint history dependent vector of the network model.

Appendix C: Construction of Network SCAQCF Measurement Model

This appendix describes the procedure to form the network SCAQCF measurement model. The task is achieved by the following two subtasks: (1) Use the mapping lists (states, equations, controls) formed in Appendix B to create the network SCAQCF measurement model from device-level SCAQCF measurement models; (2) Add the network KCL equations as additional virtual measurements to the network measurement model. The detailed procedure is illustrated below.

Step 1: Use mapping lists to form the network SCAQCF measurement model from device-level SCAQCF measurement model.

(1) Create network measurement model $Y_{m,x}$

Given:

$m_vIQDSCAQCFMeasLinTermX$ associated with $iDevice^{th}$ device
 Device State Variable to Network State Variable Mapping List:
 $pQDSCAQCFNetwork_StateIndex$

Create:

$pNetMeasModel \rightarrow m_vIQDSCAQCFMeasLinTermX$

Process:

$pDev = (Device^*)vImportedDevices[iDevice];$

Elements in $Y_{m,x}$ of the $iDevice^{th}$ device measurement model	Elements in $Y_{m,x}$ of the network measurement model
state number: $i_{dev} = pDevMeasModel \rightarrow m_vIQDSCAQCFMeasLinTermX.slinear_i;$	$pNetMeasModel \rightarrow m_vIQDSCAQCFMeasLinTermX.slinear_i = pDev \rightarrow pQDSCAQCFNetwork_StateIndex[i_{dev}];$
Coefficient: $v_{dev} = pDevMeasModel \rightarrow m_vIQDSCAQCFMeasLinTermX.slinear_v;$	$pNetMeasModel \rightarrow m_vIQDSCAQCFMeasLinTermX.slinear_v = v_{dev};$

(2) Create network measurement model Y_{m,u}

Given:

m_vlQDSCAQCFMeasLinTermU associated with *iDevice*th device
 Device Control Variable to Network Control Variable Mapping List:
 pQDSCAQCFNetwork_ControlIndex

Create:

pNetMeasModel-> m_vlQDSCAQCFMeasLinTermU

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Y _{m,u} of the <i>iDevice</i> th device measurement model	Elements in Y _{m,u} of the network measurement model
Control number: $i_{dev} = \text{pDevMeasModel->m_vlQDSCAQCFMeasLinTermU.slinear_i};$	pNetMeasModel->m_vlQDSCAQCFMeasLinTermU.slinear_i = pDev->pQDSCAQCFNetwork_ControlIndex[i_{dev}];
Coefficient: $v_{dev} = \text{pDevMeasModel->m_vlQDSCAQCFMeasLinTermU.slinear_v};$	pNetMeasModel->m_vlQDSCAQCFMeasLinTermU.slinear_v = v_{dev} ;

(3) Create network measurement model F_{m,x}

Given:

m_vqQDSCAQCFMeasNonlinTermX associated with *iDevice*th device
 Device State Variable to Network State Variable Mapping List:
 pQDSCAQCFNetwork_StateIndex

Create:

pNetMeasModel-> m_vqQDSCAQCFMeasNonlinTermX

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in F _{m,x} of the <i>iDevice</i> th device measurement model	Elements in F _{m,x} of the network measurement model
State number: $i_{dev} = \text{pDevMeasModel->m_vqQDSCAQCFMeasNonlinTermX.slinear_i};$	pNetMeasModel->m_vqQDSCAQCFMeasNonlinTermX.slinear_i = i_{dev} ;

>m_vqQDSCAQCFMeasNonlinTermX.squad_ _i;	>m_vqQDSCAQCFMeasNonlinTermX.squad _i = pDev- >pQDSCAQCFNetwork_StateIndex[i_{dev}]
State number: $j_{dev} =$ pDevMeasModel- >m_vqQDSCAQCFMeasNonlinTermX.squad_ _j;	pNetMeasModel- >m_vqQDSCAQCFMeasNonlinTermX.squad _j = pDev- >pQDSCAQCFNetwork_StateIndex[j_{dev}]
Coefficient: $v_{dev} =$ pDevMeasModel- >m_vqQDSCAQCFMeasNonlinTermX.squad_ _v;	pNetMeasModel- >m_vqQDSCAQCFMeasNonlinTermX.squad _v = v_{dev}

(4) Create network measurement model F_{m,u}

Given:

m_vqQDSCAQCFMeasNonlinTermU associated with $iDevice^{th}$ device
Device Control Variable to Network Control Variable Mapping List:
pQDSCAQCFNetwork_ControlIndex

Create:

pNetMeasModel-> m_vqQDSCAQCFMeasNonlinTermU

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in F _{m,u} of the $iDevice^{th}$ device measurement model	Elements in F _{m,u} of the network measurement model
Control number: $i_{dev} =$ pDevMeasModel- >m_vqQDSCAQCFMeasNonlinTermU.squad _i	pNetMeasModel- >m_vqQDSCAQCFMeasNonlinTermU.squad _i = pDev- >pQDSCAQCFNetwork_ControlIndex[i_{dev}]
Control number: $j_{dev} =$ pDevMeasModel- >m_vqQDSCAQCFMeasNonlinTermU.squad _j	pNetMeasModel- >m_vqQDSCAQCFMeasNonlinTermU.squad _j = pDev- >pQDSCAQCFNetwork_ControlIndex[j_{dev}]
Coefficient: $v_{dev} =$ pDevMeasModel- >m_vqQDSCAQCFMeasNonlinTermU.squad _v	pNetMeasModel- >m_vqQDSCAQCFMeasNonlinTermU.squad _v = v_{dev}

(5) Create network measurement model Fm,ux

Given:

m_vqQDSCAQCFMeasNonlinTermUX associated with $iDevice^{th}$ device

Device Control Variable to Network Control Variable Mapping List:

pQDSCAQCFNetwork_ControlIndex

Device State Variable to Network State Variable Mapping List:

pQDSCAQCFNetwork_StateIndex

Create:

pNetMeasModel-> m_vqQDSCAQCFMeasNonlinTermUX

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Fm,ux of the $iDevice^{th}$ device measurement model	Elements in Fm,ux of the network measurement model
Control number: $i_{dev} = pDevMeasModel->m_vqQDSCAQCFMeasNonlinTermUX.squad_i$	pNetMeasModel->m_vqQDSCAQCFMeasNonlinTermUX.squad_i = pDev->pQDSCAQCFNetwork_ControlIndex[i_{dev}];
State number: $j_{dev} = pDevMeasModel->m_vqQDSCAQCFMeasNonlinTermUX.squad_j$	pNetMeasModel->m_vqQDSCAQCFMeasNonlinTermUX.squad_j = pDev->pQDSCAQCFNetwork_StateIndex[j_{dev}];
Coefficient: $v_{dev} = pDevMeasModel->m_vqQDSCAQCFMeasNonlinTermUX.squad_v$	pNetMeasModel->m_vqQDSCAQCFMeasNonlinTermUX.squad_v = v_{dev} ;

(6) Create Network Measurement Model Nm,x

Given:

m_vlQDSCAQCFMeasPastTermX associated with $iDevice^{th}$ device

Device State Variable to Network State Variable Mapping List:

pQDSCAQCFNetwork_StateIndex

Create:

pNetMeasModel-> m_vIQDSCAQCFMeasPastTermX

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Nm,x of the <i>iDevice</i> th device measurement model	Elements in Nm,x of the network measurement model
State number: $i_{dev} = \text{pDevMeasModel->m_vIQDSCAQCFMeasPastTermX.slinear_i}$	pNetMeasModel->m_vIQDSCAQCFMeasPastTermX.slinear_i = pDev-> pQDSCAQCFNetwork_StateIndex[i_{dev}]
Coefficient: $v_{dev} = \text{pDevMeasModel->m_vIQDSCAQCFMeasPastTermX.slinear_v}$	pNetMeasModel->m_vIQDSCAQCFMeasPastTermX.slinear_v = v_{dev}

(7) Create Network Measurement Model Nm,u

Given:

m_vIQDSCAQCFMeasPastTermU associated with *iDevice*th device
Device Control Variable to Network Control Variable Mapping List:
pQDSCAQCFNetwork_ControlIndex

Create:

pNetMeasModel-> m_vIQDSCAQCFMeasPastTermU

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Nm,u of the <i>iDevice</i> th device measurement model	Elements in Nm,u of the network measurement model
Control number: $i_{dev} = \text{pDevMeasModel->m_vIQDSCAQCFMeasPastTermU.slinear_i}$	pNetMeasModel->m_vIQDSCAQCFMeasPastTermU.slinear_i = pDev-> pQDSCAQCFNetwork_ControlIndex[i_{dev}]
Coefficient: $v_{dev} = \text{pDevMeasModel->m_vIQDSCAQCFMeasPastTermU.slinear_v}$	pNetMeasModel->m_vIQDSCAQCFMeasPastTermU.slinear_v = v_{dev}

(8) Create Network Measurement Model Mm

Given:

m_vlQDSCAQCFMeasPastTermI associated with $iDevice^{th}$ device
 Device Equation to Network Equation Mapping List: pQDSCAQCFNetwork_EquIndex

Create:

pNetMeasModel-> m_vlQDSCAQCFMeasPastTermI

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Mm of the $iDevice^{th}$ device measurement model	Elements in Mm of the network measurement model
Equation number: $i_{dev} = pDevMeasModel->m_vlQDSCAQCFMeasPastTermI.slinear_i$	pNetMeasModel->m_vlQDSCAQCFMeasPastTermI.slinear_i = pDev-> pQDSCAQCFNetwork_EquIndex[i_{dev}]
Coefficient: $v_{dev} = pDevMeasModel->m_vlQDSCAQCFMeasPastTermI.slinear_v$	pNetMeasModel->m_vlQDSCAQCFMeasPastTermI.slinear_v = v_{dev}

(9) Create Network Measurement Model Km

Given:

m_vdQDSCAQCFMeasConstantK associated with $iDevice^{th}$ device

Create:

pNetMeasModel-> m_vdQDSCAQCFMeasConstantK

Process:

pDev=(Device*)vImportedDevices[iDevice];

Elements in Km of the $iDevice^{th}$ device measurement model	Elements in Km of the network measurement model
Coefficient: $v_{dev} = pDevMeasModel->m_vdQDSCAQCFMeasConstantK.slinear_v$	pNetMeasModel->m_vdQDSCAQCFMeasConstantK.slinear_v = v_{dev}

(10) vNetworkMeasModel.push_back(pNetMeasModel);

Step 2: Add the network KCL equations as additional virtual measurements to the network measurement model.

(1) Determine the network KCL equations and their equation numbers in the whole network model.

The network SCAQCF model can be categorized into three equation sets. The first equation set is the terminal through variable equation set. The second equation set contains the network internal KCL equations. The third equation set consists of all the device internal equations. Therefore, it's simple to locate the network KCL equations in the network model.

(2) Add these network KCL internal equations into the network measurement model.

For the k th equation (network KCL equation) in the network model:

(a) Add Yeqx to the network measurement model;

Create:

pNetMeasModel-> m_vlQDSCAQCFMeasLinTermX

Process:

Elements in Yeqx of the network model	Elements in Ym,x of the network measurement model
state number: i	pNetMeasModel->m_vlQDSCAQCFMeasLinTermX.slinear_i = i ;
Coefficient: $v =$ pQDSCAQCFNetwork_Yeqx[k][i];	pNetMeasModel->m_vlQDSCAQCFMeasLinTermX.slinear_v = v ;

(b) Add Yequ to the network measurement model;

Create:

pNetMeasModel-> m_vlQDSCAQCFMeasLinTermU

Process:

Elements in Yequ of the network model	Elements in Ym,u of the network measurement model
---------------------------------------	---

control number: i	pNetMeasModel- >m_vlQDSCAQCFMeasLinTermU.slinear_i = i ;
Coefficient: $v =$ pQDSCAQCFNetwork_Yequ[k][i];	pNetMeasModel- >m_vlQDSCAQCFMeasLinTermU.slinear_v = v ;

(c) Add Feqxx to the network measurement model;

Create:

pNetMeasModel-> m_vqQDSCAQCFMeasNonLinTermX

Process:

Elements in Feqxx of the network model	Elements in Fm,x of the network measurement model
state number: i	pNetMeasModel- >m_vqQDSCAQCFMeasNonLinTermX.squad_i = i ;
state number: j	pNetMeasModel- >m_vqQDSCAQCFMeasNonLinTermX.squad_j = j ;
Coefficient: v	pNetMeasModel- >m_vqQDSCAQCFMeasNonLinTermX.squad_v = v ;

(d) Add Fequu to the network measurement model;

Create:

pNetMeasModel-> m_vqQDSCAQCFMeasNonLinTermU

Process:

Elements in Fequu of the network model	Elements in Fm,u of the network measurement model
control number: i	pNetMeasModel- >m_vqQDSCAQCFMeasNonLinTermU.squad_i = i ;
control number: j	pNetMeasModel- >m_vqQDSCAQCFMeasNonLinTermU.squad_j = j ;

Coefficient: v	pNetMeasModel- >m_vqQDSCAQCFMeasNonLinTermU.squad_v = v ;
------------------	---

(e) Add Fequx to the network measurement model;

Create:

pNetMeasModel-> m_vqQDSCAQCFMeasNonLinTermUX

Process:

Elements in Fequx of the network model	Elements in Fm,ux of the network measurement model
state number: i	pNetMeasModel- >m_vqQDSCAQCFMeasNonLinTermUX.squad_i = i ;
control number: j	pNetMeasModel- >m_vqQDSCAQCFMeasNonLinTermUX.squad_j = j ;
Coefficient: v	pNetMeasModel- >m_vqQDSCAQCFMeasNonLinTermUX.squad_v = v ;

(f) Add Neqx to the network measurement model;

Create:

pNetMeasModel-> m_vlQDSCAQCFMeasPastTermX

Process:

Elements in Neqx of the network model	Elements in Nm,x of the network measurement model
state number: i	pNetMeasModel- >m_vlQDSCAQCFMeasPastTermX.slinear_i = i ;
Coefficient: $v =$ pQDSCAQCFNetwork_Neqx[k][i];	pNetMeasModel- >m_vlQDSCAQCFMeasPastTermX.slinear_v = v ;

(g) Add Nequ to the network measurement model;

Create:

pNetMeasModel-> m_vlQDSCAQCFMeasPastTermU

Process:

Elements in Nequ of the network model	Elements in Nm,u of the network measurement model
control number: i	pNetMeasModel- >m_vlQDSCAQCFMeasPastTermU.slinear_i = i ;
Coefficient: $v =$ pQDSCAQCFNetwork_Nequ[k][i];	pNetMeasModel- >m_vlQDSCAQCFMeasPastTermU.slinear_v = v ;

(h) Add Meq to the network measurement model;

Create:

pNetMeasModel-> m_vlQDSCAQCFMeasPastTermI

Process:

Elements in Meq of the network model	Elements in Mm of the network measurement model
through variable number: i	pNetMeasModel- >m_vlQDSCAQCFMeasPastTermI.slinear_i = i ;
Coefficient: $v =$ pQDSCAQCFNetwork_Meq[k][i];	pNetMeasModel- >m_vlQDSCAQCFMeasPastTermI.slinear_v = v ;

(i) Add Keq to the measurement model;

Create:

pNetMeasModel-> m_vdQDSCAQCFMeasConstantK

Process:

Elements in Keq of the network model	Elements in Nm,x of the network measurement model
Coefficient: $v =$ pQDSCAQCFNetwork_Keqx[k];	pNetMeasModel- >m_vdQDSCAQCFMeasConstantK = v ;

The final expression for the network SCAQCF measurement model is:

$$\mathbf{z} = Y_{netm,x}\mathbf{x}(t) + Y_{netm,u}\mathbf{u}(t) + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{x}^T F_{netm,x}^i \mathbf{x} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T F_{netm,u}^i \mathbf{u} \\ \mathbf{M} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{M} \\ \mathbf{u}^T F_{netm,ux}^i \mathbf{x} \\ \mathbf{M} \end{array} \right\} + C_{netm} + \boldsymbol{\eta}$$

$$C_{netm} = N_{netm,x}\mathbf{x}(t-h) + N_{netm,u}\mathbf{u}(t-h) + M_{netm}\mathbf{i}(t-h) + K_{netm}$$

Measurement noise error: dMeterScale, dMeterSigmaPU

Note: All the above variables are in metric system.

where:

\mathbf{z} : measurement variables at both time t and time t_m , $\mathbf{z} = [\mathbf{z}(t), \mathbf{z}(t_m)]$;

\mathbf{x} : external and internal state variables of the measurement model, $\mathbf{x} = [\mathbf{x}(t), \mathbf{x}(t_m)]$;

\mathbf{u} : control variables of the measurement model, i.e. transformer tap, etc. $\mathbf{u} = [\mathbf{u}(t), \mathbf{u}(t_m)]$;

$Y_{netm,x}$: matrix defining the linear part for state variables of the network measurement model;

$F_{netm,x}$: matrices defining the quadratic part for state variables of the network measurement model;

$Y_{netm,u}$: matrix defining the linear part for control variables of the network measurement model;

$F_{netm,u}$: matrices defining the quadratic part for control variables of the network measurement model;

$F_{netm,xu}$: matrices defining the quadratic part for the product of state and control variables of the network measurement model;

C_{netm} : history dependent vector of the network measurement model;

$N_{netm,x}$: matrix defining the last integration step state variables part of the network measurement model;

$N_{netm,u}$: matrix defining the last integration step control variables part of the network measurement model;

M_{netm} : matrix defining the last integration step through variables part of the network measurement model;

K_{netm} : constant vector of the network measurement model;

$dMeterScale$: the scale that meters use (in metric units);

$dMeterSigmaPU$: the standard deviation for the measurements (in per. unit).

Appendix D: Device Operating Constraints

This appendix lists some typical devices in the distribution system and illustrates the operating constraints of these devices. As described in Section 3.2, the device operating constraints are stored in $\mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}_{\max}$, and the formation algorithm of network operating constraints (introduced in Section 4.3) works directly on these device-level operating constraints.

Operating Constraints of Distribution Lines and Single-Phase Circuits

The operating constraint of a distribution line or a single-phase circuit is that the current phasor magnitude shall be less than the ampacity of the line conductor at each phase. The general expression for the operating constraint using variables in Figure D.1 is:

$$0 \leq \left| \mathcal{I}_{ABCN}^{\%} \right| \leq I_{ampacity}, \text{ and} \quad (\text{D.1})$$

$$0 \leq \left| \mathcal{I}_{abcn}^{\%} \right| \leq I_{ampacity}, \quad (\text{D.2})$$

where $\mathcal{I}_{ABCN}^{\%}$ denotes the current phasors at one terminal of the line, $\mathcal{I}_{abcn}^{\%}$ denotes the current phasors at the other terminal of the line, and $I_{ampacity}$ is the ampacity of the line conductor.

The quadratic form of the operating constraint is:

$$\begin{cases} 0 \leq I_{ABCN,real}^2 + I_{ABCN,imag}^2 \leq I_{ampacity}^2 \\ 0 \leq I_{abcn,real}^2 + I_{abcn,imag}^2 \leq I_{ampacity}^2 \end{cases} \Rightarrow 0 \leq \sum_i \sum_j Y_{eqx,i}^{2k} Y_{eqx,j}^{2k} x_i x_j + \sum_i \sum_j Y_{eqx,i}^{2k+1} Y_{eqx,j}^{2k+1} x_i x_j \leq I_{ampacity}^2, \quad (\text{D.3})$$

where $I_{ABCN,real}$ and $I_{ABCN,imag}$ are the real and imaginary part of the current phasors at one terminal of the line, $I_{abcn,real}$ and $I_{abcn,imag}$ are the real and imaginary part of the current phasors at the other terminal of the line, the superscript k represents the terminal k of the SCAQCF device model, $Y_{eqx,i}^{2k}$ represents the i -th element of the $2k$ -th row of the matrix Y_{eqx} , $Y_{eqx,j}^{2k}$ represents the j -th element of the $2k$ -th row of the matrix Y_{eqx} , $Y_{eqx,i}^{2k+1}$ represents the i -th element of the $(2k+1)$ -th row of the matrix Y_{eqx} , $Y_{eqx,j}^{2k+1}$ represents the j -th element of the $(2k+1)$ -th row of the matrix Y_{eqx} , x_i is the i -th state of the device, and x_j is the j -th state of the device.

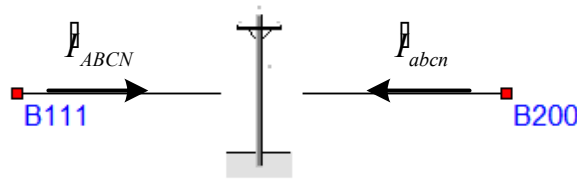


Figure D.1: Distribution Line

Operating Constraints of Residential Loads

The operating constraint of a residential load is that the voltage magnitude shall be larger than 0.95 p.u. but less than 1.05 p.u. The load model is shown in Figure D.2. The general expression of the operating constraint is:

$$\begin{aligned} 0.95V_{rated} &\leq |V_{L1}^{\%}| \leq 1.05V_{rated} \\ 0.95V_{rated} &\leq |V_{L2}^{\%}| \leq 1.05V_{rated} \end{aligned} \quad (D.4)$$

where

$$\begin{aligned} |V_{L1}^{\%}| &= \sqrt{V_{L1,real}^2 + V_{L1,imag}^2} \\ |V_{L2}^{\%}| &= \sqrt{V_{L2,real}^2 + V_{L2,imag}^2} \end{aligned} \quad (D.5)$$

The quadratic form of the operating constraint is:

$$\begin{aligned} 0.95V_{rated}^2 &\leq V_{L1,real}^2 + V_{L1,imag}^2 \leq 1.05V_{rated}^2 \\ 0.95V_{rated}^2 &\leq V_{L2,real}^2 + V_{L2,imag}^2 \leq 1.05V_{rated}^2 \end{aligned} \quad (D.6)$$

where $V_{L1}^{\%}$ and $V_{L2}^{\%}$ are the voltage phasors at L1 and L2, $V_{L1,real}$ and $V_{L1,imag}$ are the real and imaginary parts of $V_{L1}^{\%}$, $V_{L2,real}$ and $V_{L2,imag}$ are the real and imaginary parts of $V_{L2}^{\%}$, and V_{rated} is the rated voltage of the load.

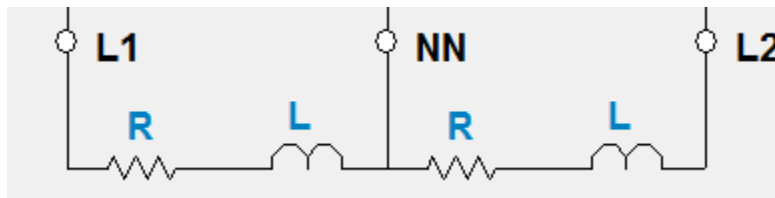


Figure D.2: Balanced Load at the Secondary Bus (Residential Load)

Operating Constraint of a Transformer with Secondary Center Tap

The operating constraint of a transformer with the secondary center tap is that the magnitude of the current phasor through the primary side shall be less than the rated current value. The general expression for the operating constraint using variables in Figure D.3 is:

$$0 \leq |I_0^{\%}| \leq I_{rated} \quad (D.7)$$

where I_0^c is the current phasor through the primary side of the transformer, I_{rated} is the rated current at the primary side of the transformer.

The quadratic form of the operating constraint is:

$$0 \leq I_{0,real}^2 + I_{0,imag}^2 \leq I_{rated}^2 \Rightarrow 0 \leq \sum_i \sum_j Y_{eqx,i}^0 Y_{eqx,j}^0 x_i x_j + \sum_i \sum_j Y_{eqx,i}^1 Y_{eqx,j}^1 x_i x_j \leq I_{rated}^2, \quad (D.8)$$

where $I_{0,real}$ and $I_{0,imag}$ are the real and imaginary parts of phasor I_0^c , $Y_{eqx,i}^0$ represents the i -th element of the first row of the matrix Y_{eqx} , $Y_{eqx,j}^0$ represents the j -th element of the first row of the matrix Y_{eqx} , $Y_{eqx,i}^1$ represents the i -th element of the second row of the matrix Y_{eqx} , $Y_{eqx,j}^1$ represents the j -th element of the second row of the matrix Y_{eqx} , x_i is the i -th state of the device, and x_j is the j -th state of the device.

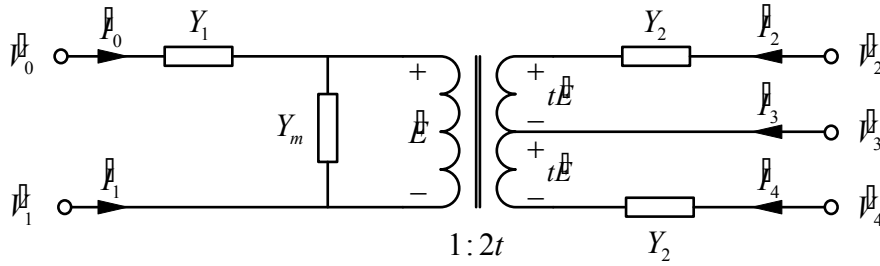


Figure D.3: Transformer with Secondary Center Tap

Operating Constraint of the Battery

The operating constraints of the battery are: (1) the DC current shall be less than the ampacity of the battery; (2) the state of charge (SOC) of the battery shall be in a proper range. The general expression is:

$$-I_{DC,max} \leq I_{ADr} \leq I_{DC,max}, \quad \text{and} \quad (D.9)$$

$$0.1SOC_{max} \leq SOC \leq 1.0SOC_{max}, \quad (D.10)$$

where I_{ADr} is the current output of the battery, $I_{DC,max}$ is the maximum output current, SOC_{max} is the maximum charge the battery can hold, and the unit of SOC_{max} is kilo-Coulombs.

Since the operating constraint of the battery is linear, the quadratic expression of the operating constraint is the same as the operating constraint listed above.

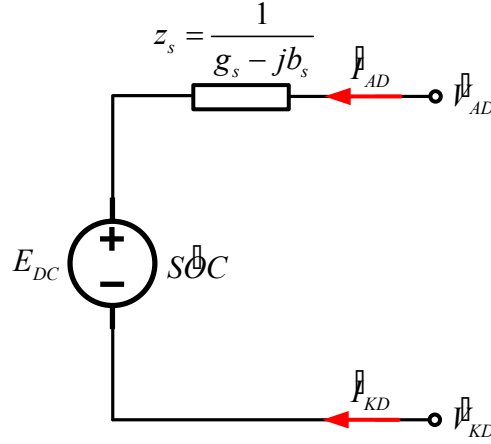


Figure D.4: Battery

Operating Constraint of the Converter

The operating constraints of the converter are: (1) both the DC and AC currents shall be less than the corresponding rated current value; (2) the modulation index of the converter shall be in a proper range; (3) the output real and imaginary power shall be less than the rated power.

The general expression of these constraints is:

$$0 \leq I_{ADr} \leq I_{DC, rated} \quad (D.11)$$

$$0 \leq \left| \frac{P}{abc} \right| \leq I_{AC, rated} \quad (D.12)$$

$$0 \leq m \leq 1.0 \quad (D.13)$$

$$0.0 \leq P_{ac}^2 + Q_{ac}^2 \leq S_{rated}^2 \quad (D.14)$$

$$0.0 \leq P_{ref}^2 + Q_{ref}^2 \leq S_{rated}^2 \quad (D.15)$$

where I_{ADr} is the DC current of the converter, $I_{DC, rated}$ is the rated DC current value of the converter, $\frac{P}{abc}$ denotes the current phasors at AC side of the converter, $I_{AC, rated}$ is the rated AC current value of the converter, m is the modulation index of the SPWM converter, P_{ac} and Q_{ac} are the output real and reactive power, P_{ref} and Q_{ref} are the desired output active and reactive power of the converter, and S_{rated} is the rated power of the converter.

The quadratic form of the operating constraint is:

$$0 \leq I_{ADr} \leq I_{DC, max} \quad (D.16)$$

$$0 \leq I_{real}^2 + I_{imag}^2 \leq I_{rated}^2 \Rightarrow 0 \leq \sum_i \sum_j Y_{eqx,i}^{2k} Y_{eqx,j}^{2k} x_i x_j + \sum_i \sum_j Y_{eqx,i}^{2k+1} Y_{eqx,j}^{2k+1} x_i x_j \leq I_{rated}^2 \quad (D.17)$$

$$0 \leq m \leq 1.0, \quad (\text{D.18})$$

$$0.0 \leq P_{ac}^2 + Q_{ac}^2 \leq S_{rated}^2, \quad (\text{D.19})$$

$$0.0 \leq P_{ref}^2 + Q_{ref}^2 \leq S_{rated}^2, \quad (\text{D.20})$$

where the superscript k represents the terminal k of the SCAQCF device model, $Y_{eqx,i}^{2k}$ represents the i -th element of the $2k$ -th row of the matrix Y_{eqx} , $Y_{eqx,j}^{2k}$ represents the j -th element of the $2k$ -th row of the matrix Y_{eqx} , $Y_{eqx,i}^{2k+1}$ represents the i -th element of the $2k+1$ -th row of the matrix Y_{eqx} , $Y_{eqx,j}^{2k+1}$ represents the j -th element of the $2k+1$ -th row of the matrix Y_{eqx} , x_i is the i -th state of the device, and x_j is the j -th state of the device.

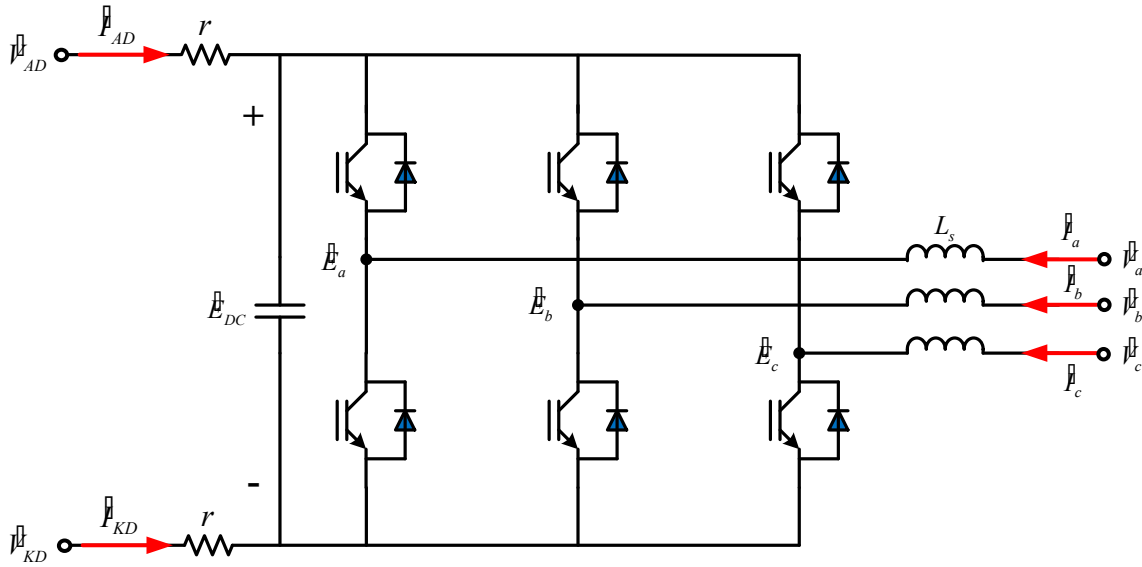


Figure D.5: Converter

Appendix E: Construction of Equality Constraints in Quadratized OPF Problem

This appendix introduces the procedure to form the equality constraints in the quadratized OPF problem from the network SCAQCF model. The equality constraints are defined in section 5.1, and the network SCAQCF model is defined in section 4.3. The problem is stated as follows. Given the network SCAQCF model, we compute the equality constraints. The end result is stored in the following arrays.

double pEquConstraint_Yeqx;**

Dimension: nQDSCAQCFNetwork_Equ by nQDSCAQCFNetwork_State;
Coefficients of the linear state variables for the equality constraints.
It stores the entry's row number, column number and entry value.

double pEquConstraint_Yequ;**

Dimension: pQDSCAQCFNetwork_Equ by nQDSCAQCFNetwork_Control;
Coefficients of the linear control variables for the equality constraints.
It stores the entry's row number, column number and entry value.

SP_CUBIX* pEquConstraint_Feqxx;

Dimension: SP_CUBIX[nQDSCAQCFNetwork_Feqxx]
Coefficients of the quadratic state variables for the equality constraints.
It stores the entry's row number, column number, entry value and its position in the quadratic term.

SP_CUBIX* pEquConstraint_Fequu;

Dimension: SP_CUBIX[nQDSCAQCFNetwork_Fequu];
Coefficients of the quadratic control variables for the equality constraints.
It stores the entry's row number, column number, entry value and its position in the quadratic term.

SP_CUBIX* pEquConstraint_Fequx;

Dimension: SP_CUBIX[nQDSCAQCFNetwork_Fequx]
Coefficients of the production of state and control variables for the equality constraints.
It stores the entry's row number, column number, entry value and its position in the quadratic term.

double pEquConstraint_Neqx;**

Dimension: nQDSCAQCFNetwork_Equ by nQDSCAQCFNetwork_StateOver2;
Coefficients of the past state variables for the equality constraints.

It stores the entry's row number, column number and entry value.

double pEquConstraint_Nequ;**

Dimension: nQDSCAQCFNetwork_Equ by nQDSCAQCFNetwork_ControlOver2;

Coefficients of the past control variables for the equality constraints.

It stores the entry's row number, column number and entry value.

double pEquConstraint_Meq;**

Dimension: nQDSCAQCFNetwork_Equ by nQDSCAQCFNetwork_Node * 2;

Coefficients of the past through variables for the equality constraints.

It stores the entry's row number, column number and entry value.

double* pEquConstraint_Keq;

Dimension: nQDSCAQCFNetwork_Equ by 1;

Constant vectors for the equality constraints.

It stores the entry's row number and entry value.

CString* pEquConstraint_NodeName;

Dimension: nQDSCAQCFNetwork_Node by 1;

It stores the network node names.

double* pEquConstraint_StateNormFactor;

Dimension: nQDSCAQCFNetwork_State by 1;

Normalization factors of the states.

It stores the entry's row number and entry value.

double* pEquConstraint_ThroughNormFactor;

Dimension: nQDSCAQCFNetwork_Equ by 1;

Normalization factors of the through variables.

It stores the entry's row number and entry value.

double* pEquConstraint_ControlNormFactor;

Dimension: nQDSCAQCFNetwork_Control by 1;

Normalization factors of the control variables.

It stores the entry's row number and entry value.

The array formation procedure for this subsection is simple. As a matter of fact, the arrays defined in this subsection are copied from the corresponding arrays of the network.

Equality Constraints Arrays Formation

- (1) pEquConstraint_Reqx is copied from pQDSCAQCFNetwork_Reqx (Y_{eqx}) of the network model;
- (2) pEquConstraint_Requ is copied from pQDSCAQCFNetwork_Requ (Y_{equ}) of the network model;
- (3) pEquConstraint_Feqxx is copied from pQDSCAQCFNetwork_Feqxx (F_{eqxx}) of the network model;
- (4) pEquConstraint_Fequu is copied from pQDSCAQCFNetwork_Fequu (F_{equu}) of the network model;
- (5) pEquConstraint_Fequx is copied from pQDSCAQCFNetwork_Fequx (F_{equx}) of the network model;
- (6) pEquConstraint_Neqx is copied from pQDSCAQCFNetwork_Neqx (N_{eqx}) of the network model;
- (7) pEquConstraint_Nequ is copied from pQDSCAQCFNetwork_Nequ (N_{equ}) of the network model;
- (8) pEquConstraint_Meq is copied from pQDSCAQCFNetwork_Meq (M_{eq}) of the network model;
- (9) pEquConstraint_Keq is copied from pQDSCAQCFNetwork_Keq (K_{eq}) of the network model;
- (10) pEquConstraint_NodeName is copied from pQDSCAQCFNetwork_NodeName of the network model;
- (11) pEquConstraint_StateNormFactor is copied from pQDSCAQCFNetwork_StateNormFactor of the network model;
- (12) pEquConstraint_ThroughNormFactor is copied from pQDSCAQCFNetwork_ThroughNormFactor of the network model;
- (13) pEquConstraint_ControlNormFactor is copied from pQDSCAQCFNetwork_ControlNormFactor of the network model.

Appendix F: Construction of Inequality Constraints in Quadratized OPF Problem

This appendix introduces the procedure to form the inequality constraints in the quadratized OPF problem from the network SCAQCF model. The inequality constraints are defined in section 5.2, and the network SCAQCF model is defined in section 4.3. The problem is stated as follows. Given the network SCAQCF model, we compute the inequality constraints. The end result is stored in the following arrays.

double pInEquConstraint_Yeqx;**

Dimension: $nQDSCAQCFNetwork_Constraint * 2$ by $nQDSCAQCFNetwork_State$;
Coefficients of the linear state variables for the inequality constraints.
It stores the entry's row number, column number and entry value.

double pInEquConstraint_Yequ;**

Dimension: $nQDSCAQCFNetwork_Constraint * 2$ by $nQDSCAQCFNetwork_Control$;
Coefficients of the linear control variables for the inequality constraints.
It stores the entry's row number, column number and entry value.

SP_CUBIX* pInEquConstraint_Feqxx;

Dimension: $SP_CUBIX[nQDSCAQCFNetwork_Ffeqxx * 2]$;
Coefficients of the quadratic state variables for the inequality constraints.
It stores the entry's row number, column number, entry value and its position in the quadratic term.

SP_CUBIX* pInEquConstraint_Fequu;

Dimension: $SP_CUBIX[nQDSCAQCFNetwork_Ffequ * 2]$
Coefficients of the quadratic control variables for the inequality constraints.
It stores the entry's row number, column number, entry value and its position in the quadratic term.

SP_CUBIX* pInEquConstraint_Fequx;

Dimension: $SP_CUBIX[nQDSCAQCFNetwork_Ffequx * 2]$
Coefficients of the production of state and control variables for the inequality constraints.
It stores the entry's row number, column number, entry value and its position in the quadratic term.

double* pInEquConstraint_Ceqc;

Dimension: $nQDSCAQCFNetwork_Constraint * 2$ by 1;
History dependent vectors for the inequality constraints.

It stores the entry's row number and entry value.

The arrays are formed as follows.

(1) Formulate pInEquConstraint_Yeqx;

Elements in Yfeqx in the network model	Elements in pInEquConstraint_Yeqx
constraint equation number: i_{net}	$\begin{aligned} \text{pInEquConstraint_Yeqx}[i_{net}][j_{net}] &= -v_{dev}; \\ \text{pInEquConstraint_Yeqx}[i_{net} + \text{nQDSCAQCFNetwork_Constraint}][j_{net}] &= v_{dev}; \end{aligned}$
state number: j_{net}	
coefficient: $v_{net} = \text{pQDSCAQCFNetwork_Yfeqx}[i_{net}][j_{net}]$	

(2) Formulate pInEquConstraint_Yequ;

Elements in Yfequ in the network model	Elements in pInEquConstraint_Yequ
constraint equation number: i_{net}	$\begin{aligned} \text{pInEquConstraint_Yequ}[i_{net}][j_{net}] &= -v_{dev}; \\ \text{pInEquConstraint_Yequ}[i_{net} + \text{nQDSCAQCFNetwork_Constraint}][j_{net}] &= v_{dev}; \end{aligned}$
control number: j_{net}	
coefficient: $v_{net} = \text{pQDSCAQCFNetwork_Yfequ}[i_{net}][j_{net}]$	

(3) Formulate pInEquConstraint_Feqxx;

Elements in Ffeqxx in the network model	Elements in pInEquConstraint_Feqxx
constraint equation number: k_{net}	$\text{pInEquConstraint_Feqxx}[i_{\text{Feqxx}}].\text{scubix_k} = k_{net};$
	$\text{pInEquConstraint_Feqxx}[i_{\text{Feqxx}} + \text{nQDSCAQCFNetwork_Ffeqxx}].\text{scubix_k} = k_{net} + \text{nQDSCAQCFNetwork_Constraint};$
state number: i_{net}	$\text{pInEquConstraint_Feqxx}[i_{\text{Feqxx}}].\text{scubix_i} = i_{net};$

	$\text{pInEquConstraint_Feqxx}[i\text{Feqxx} + \text{nQDSCAQCFNetwork_Ffeqxx}].\text{scubix_i} = i_{net};$
state number: j_{net}	$\text{pInEquConstraint_Feqxx}[i\text{Feqxx}].\text{scubix_j} = j_{net};$
	$\text{pInEquConstraint_Feqxx}[i\text{Feqxx} + \text{nQDSCAQCFNetwork_Ffeqxx}].\text{scubix_j} = j_{net};$
Coefficient: v_{net}	$\text{pInEquConstraint_Feqxx}[i\text{Feqxx}].\text{scubix_v} = -v_{net};$
	$\text{pInEquConstraint_Feqxx}[i\text{Feqxx} + \text{nQDSCAQCFNetwork_Ffeqxx}].\text{scubix_v} = v_{net};$ $i\text{Feqxx}++;$

(4) Formulate pInEquConstraint_Fequu;

Elements in Ffequ in the network model	Elements in pInEquConstraint_Fequu
constraint equation number: k_{net}	$\text{pInEquConstraint_Fequu}[i\text{Fequu}].\text{scubix_k} = k_{net};$
	$\text{pInEquConstraint_Fequu}[i\text{Fequu} + \text{nQDSCAQCFNetwork_Ffequu}].\text{scubix_k} = k_{net} + \text{nQDSCAQCFNetwork_Constraint};$
control number: i_{net}	$\text{pInEquConstraint_Fequu}[i\text{Fequu}].\text{scubix_i} = i_{net};$
	$\text{pInEquConstraint_Fequu}[i\text{Fequu} + \text{nQDSCAQCFNetwork_Ffequu}].\text{scubix_i} = i_{net};$
control number: j_{net}	$\text{pInEquConstraint_Fequu}[i\text{Fequu}].\text{scubix_j} = j_{net};$
	$\text{pInEquConstraint_Fequu}[i\text{Fequu} + \text{nQDSCAQCFNetwork_Ffequu}].\text{scubix_j} = j_{net};$
Coefficient: v_{net}	$\text{pInEquConstraint_Fequu}[i\text{Fequu}].\text{scubix_v} = -v_{net};$
	$\text{pInEquConstraint_Fequu}[i\text{Fequu} + \text{nQDSCAQCFNetwork_Ffequu}].\text{scubix_v} = v_{net};$ $i\text{Fequu}++;$

(5) Formulate pInEquConstraint_Feqx;

Elements in Ffeqx in the network model	Elements in pInEquConstraint_Feqx
--	-----------------------------------

constraint equation number: k_{net}	$pInEquConstraint_Fequx[iFequx].scubix_k = k_{net};$
	$pInEquConstraint_Fequx[iFequx + nQDSCAQCFNetwork_Ffequx].scubix_k = k_{net} + nQDSCAQCFNetwork_Constraint;$
control number: i_{net}	$pInEquConstraint_Fequx[iFequx].scubix_i = i_{net};$
	$pInEquConstraint_Fequx[iFequx + nQDSCAQCFNetwork_Ffequx].scubix_i = i_{net};$
state number: j_{net}	$pInEquConstraint_Fequx[iFequx].scubix_j = j_{net};$
	$pInEquConstraint_Fequx[iFequx + nQDSCAQCFNetwork_Ffequx].scubix_j = j_{net};$
Coefficient: v_{net}	$pInEquConstraint_Fequx[iFequx].scubix_v = -v_{net};$
	$pInEquConstraint_Fequx[iFequx + nQDSCAQCFNetwork_Ffequx].scubix_v = v_{net};$ $iFequx++;$

(6) Formulate pInEquConstraint_Ceqc;

Elements in Cfeqc in the network model	Elements in pInEquConstraint_Ceqc
constraint equation number: k_{net}	$pInEquConstraint_Ceqc[k_{net}] = -v_{net} + hmin_{net};$ $pInEquConstraint_Ceqc[k_{net} + nQDSCAQCFNetwork_Constraint] = v_{net}; -$ $hmax_{net};$
coefficient: $v_{net} =$ $pQDSCAQCFNetwork_Cfeqc[k_{net}]$	
$hmin_{net} =$ $pQDSCAQCFNetwork_hmin[k_{net}]$	
$hmax_{net} =$ $pQDSCAQCFNetwork_hmax[k_{net}]$	

Appendix G: Construction of Control Constraints in Quadratized OPF Problem

This appendix introduces the procedure to form the control constraints in the quadratized OPF problem from the network SCAQCF model. The control constraints are defined in section 5.3, and the network SCAQCF model is defined in section 4.3. The problem is stated as follows. Given the control constraints from the network SCAQCF model, we compute the control constraints of the OPF problem. The end result is stored in the following arrays.

double pControlConstraint_umin;**
Dimension: nQDSCAQCFNetwork_Control by 1;
Minimum values for the control variables.

double pControlConstraint_umax;**
Dimension: nQDSCAQCFNetwork_Control by 1;
Maximum values for the control variables.

The array formation procedure is simple. As a matter of fact, the defined arrays are copied from the corresponding arrays of the network.

Control Constraints Arrays Formation

- (1) pControlConstraint_umin is copied from pQDSCAQCFNetwork_umin (u_{\min}) of the network model;
- (2) pControlConstraint_umax is copied from pQDSCAQCFNetwork_umax (u_{\max}) of the network model;

Appendix H: Construction of the Objective Function in Quadratized OPF Problem

This appendix introduces the procedure to form the objective function in the quadratized OPF problem from the network SCAQCF model. The objective function is defined in section 5.4, and the network SCAQCF model is defined in section 4.3. The problem is stated as follows. Given the network node name list and its network index from the network SCAQCF model, we compute the objective function of the OPF problem. The end result is stored in the following arrays.

double* pObjFunction_Yeqx;

Dimension: 1 by nQDSCAQCFNetwork_State;
Coefficients of the linear state variables in the objective function.
It stores the entry's column number and entry value.

double* pObjFunction_Yequ;

Dimension: 1 by nQDSCAQCFNetwork_Control;
Coefficients of the linear control variables in the objective function.
It stores the entry's column number and entry value.

double pObjFunction_Ceqc;

Dimension: 1 by 1;
Constant value in the objective function

double pObjFunction_Feqxx;**

Dimension: nQDSCAQCFNetwork_State by nQDSCAQCFNetwork_State;
Coefficients of the quadratic state variables in the objective function.
It stores the entry's row number, column number and entry value in the quadratic term.

double pObjFunction_Fequu;**

Dimension: nQDSCAQCFNetwork_Control by nQDSCAQCFNetwork_Control;
Coefficients of the quadratic control variables in the objective function.
It stores the entry's row number, column number and entry value in the quadratic term.

double pObjFunction_Fequx;**

Dimension: nQDSCAQCFNetwork_State by nQDSCAQCFNetwork_Control;
Coefficients of the production of state and control variables in the objective function.
It stores the entry's row number, column number and entry value in the quadratic term.

The arrays are formed as follows.

The objective function after expansion is:

$$\begin{aligned} \text{minimize: } J &= \sum_{i \in \{\text{selected nodes/phases}\}} \left(\frac{V_{i,mag} - V_{i,target}}{\alpha_i V_{i,target}} \right)^2 \\ &= \sum_{i \in \{\text{selected nodes/phases}\}} \frac{1}{\alpha_i^2 V_{i,target}^2} V_{i,mag}^2 + \sum_{i \in \{\text{selected nodes/phases}\}} \frac{-2}{\alpha_i^2 V_{i,target}} V_{i,mag} + \sum_{i \in \{\text{selected nodes/phases}\}} \frac{1}{\alpha_i^2} \end{aligned} \quad \text{Not}$$

ice that the objective function only contains coefficients of the linear state variables, coefficients of the quadratic state variables and the constant value.

$$\sum_{i \in \{\text{selected nodes/phases}\}} \frac{1}{\alpha_i^2 V_{i,target}^2} \text{ is stored in } \text{pObjFunction_Feqxx};$$

$$\sum_{i \in \{\text{selected nodes/phases}\}} \frac{-2}{\alpha_i^2 V_{i,target}} \text{ is stored in } \text{pObjFunction_Yeqx};$$

$$\sum_{i \in \{\text{selected nodes/phases}\}} \frac{1}{\alpha_i^2} \text{ is stored in } \text{pObjFunction_Ceqc}.$$

(1) Formulate pObjFunction_Yeqx;

Process:

Go through all the node names (pQDSCAQCFNetwork_NodeName) in this network, find the node names with “_MG”;

Node names with “_MG” in the network	Create pObjFunction_Yeqx
node number: i	Optimal node number in the network: $i_{net} = \text{pQDSCAQCFNetwork_OptimalNodeNumber}[i]$;
	State Number: $j_{net} = i_{net} * 2$;
	Coefficient: $v = -2 / (\alpha_{j_{net}}^2 * \text{pQDSCAQCFNetwork_StateNormFactor}[j_{net}])$;
	$\text{pObjFunction_Yeqx}[j_{net}] = v$;

(2) Formulate pObjFunction_Yeqx;

Process:

In this case, pObjFunction_Yeqx = 0;

(3) Formulate pObjFunction_Ceqc;

Process:

Go through all the node names (pQDSCAQCFNetwork_NodeName) in this network, find the node names with “_MG”;

Node names with “_MG” in the network	Create pObjFunction_Ceqc
node number: i	Optimal node number in the network: $i_{net} = \text{pQDSCAQCFNetwork_OptimalNodeNumber}[i]$;
	State Number: $j_{net} = i_{net} * 2$;
	Coefficient: $v = 1 / \alpha_{j_{net}}^2$;
	$\text{pObjFunction_Ceqc} = \text{pObjFunction_Ceqc} + v$;

(4) Formulate pObjFunction_Feqxx;

Process:

Go through all the node names (pQDSCAQCFNetwork_NodeName) in this network, find the node names with “_MG”;

Node names with “_MG” in the network	Create pObjFunction_Feqxx
node number: i	Optimal node number in the network: $i_{net} = \text{pQDSCAQCFNetwork_OptimalNodeNumber}[i]$;
	State Number: $j_{net} = i_{net} * 2$;
	Coefficient: $v = 1 / (\alpha_{j_{net}}^2 * \text{pQDSCAQCFNetwork_StateNormFactor}[j_{net}]^2)$;
	$\text{pObjFunction_Feqxx}[j_{net}][j_{net}] = v$;

(5) Formulate pObjFunction_Fequu;

Process:

In this case, $\text{pObjFunction_Fequu} = 0$;

(6) Formulate pObjFunction_Fequx;

Process:

In this case, $\text{pObjFunction_Fequx} = 0$;

Appendix I: Linearization of the Quadratized OPF Problem

This appendix introduces the linearization of the quadratized OPF problem using co-state method. The general expression of the quadratized OPF problem is described in section 5. Since the control constraints are linear and the equality constraints are already taken into account when linearizing the objective function and inequalities, this appendix will illustrate the method to linearize the objective function and inequality constraints.

I.1: Linearization of the Objective Function

Denote the present operating condition with $\mathbf{u} = \mathbf{u}^0$ and $\mathbf{x} = \mathbf{x}^0$. Linearization of the objective function $J(\mathbf{x}, \mathbf{u})$ around the present operating point, yields:

$$J(\mathbf{x}, \mathbf{u}) \cong J(\mathbf{x}^0, \mathbf{u}^0) + \frac{dJ(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}}(\mathbf{u} - \mathbf{u}^0), \quad (\text{I.1})$$

where

$$\frac{dJ(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} = \frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} + \frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{u}}. \quad (\text{I.2})$$

Since the objective function is quadratic and all the coefficient matrices have been defined and formed in Section 5, the partial derivatives $\frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}$ and $\frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}}$ can be computed directly:

$$\frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} = Y_{obju} + \{\mathbf{u}^{0,T} F_{obju}^i\} + \{F_{obju}^i \mathbf{u}^0\}^T + \{F_{objux}^i \mathbf{x}^0\}^T, \text{ and} \quad (\text{I.3})$$

$$\frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} = Y_{objx} + \{\mathbf{x}^{0,T} F_{objx}^i\} + \{F_{objx}^i \mathbf{x}^0\}^T + \{\mathbf{u}^{0,T} F_{objux}^i\}. \quad (\text{I.4})$$

The derivative $\frac{d\mathbf{x}}{d\mathbf{u}}$ is obtained from the equality constraints $g(\mathbf{x}, \mathbf{u}) = 0$. Upon differentiation of the equality constraints, with respect to control variable \mathbf{u} , we have

$$\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} + \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{u}} = 0. \quad (\text{I.5})$$

The solution of $\frac{d\mathbf{x}}{d\mathbf{u}}$ is:

$$\frac{d\mathbf{x}}{d\mathbf{u}} = - \left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \right)^{-1} \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}. \quad (\text{I.6})$$

Since the equality constraints are also quadratic and all the coefficient matrices have been defined and formed in Section 5, the partial derivatives $\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}}$ and $\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}$ are automatically computed as follows,

$$\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} = Y_{equ} + \{\mathbf{u}^{0,T} F_{equ}^i\} + \{F_{equ}^i \mathbf{u}^0\}^T + \{F_{equx}^i \mathbf{x}^0\}^T, \text{ and} \quad (\text{I.7})$$

$$\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} = Y_{eqx} + \{\mathbf{x}^{0,T} F_{eqx}^i\} + \{F_{eqx}^i \mathbf{x}^0\}^T + \{\mathbf{u}^{0,T} F_{equx}^i\}. \quad (\text{I.8})$$

Back substitution in $\frac{dJ(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}}$ yields:

$$\frac{dJ(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} = \frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} - \frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \right)^{-1} \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}. \quad (\text{I.9})$$

Note that $\frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \right)^{-1}$ is the co-state vector that is independent of the control variables, and it is pre-computed at the present operating point. The vector is represented by $\hat{\mathbf{x}}_J$, and we have:

$$\hat{\mathbf{x}}_J = \frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \right)^{-1}. \quad (\text{I.10})$$

Therefore, the linearization of the objective function around the present operating point is:

$$J(\mathbf{x}, \mathbf{u}) \cong J(\mathbf{x}^0, \mathbf{u}^0) + \left(\frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} - \hat{\mathbf{x}}_J \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} \right) (\mathbf{u} - \mathbf{u}^0). \quad (\text{I.11})$$

Substitute the current operating point $(\mathbf{x}^0, \mathbf{u}^0)$ into (I.11), we have the final expression of the linearized objective function:

$$J = \mathbf{c}^T \Delta \mathbf{u} + d_J, \quad (\text{I.12})$$

where $\Delta \mathbf{u}$ is the increment of the control variable \mathbf{u} ,

$$\Delta \mathbf{u} = \mathbf{u} - \mathbf{u}^0, \quad (\text{I.13})$$

\mathbf{c} is the linear coefficient vector of $\Delta \mathbf{u}$,

$$\mathbf{c}^T = \frac{dJ(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} = \frac{\partial J(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} - \hat{\mathbf{x}}_j \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}, \quad (\text{I.14})$$

d_j is a constant value,

$$d_j = J(\mathbf{x}^0, \mathbf{u}^0). \quad (\text{I.15})$$

I.2: Linearization of the Inequality Constraints

The inequality constraints are linearized in a similar way. Denote the present operating condition with $\mathbf{u} = \mathbf{u}^0$ and $\mathbf{x} = \mathbf{x}^0$. Linearization of $h(\mathbf{x}^0, \mathbf{u}^0)$ around the present operating point, yields:

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) \cong \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0) + \frac{d\mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} (\mathbf{u} - \mathbf{u}^0), \quad (\text{I.16})$$

where

$$\frac{d\mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} = \frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} + \frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{u}}. \quad (\text{I.17})$$

Since the inequality constraints are also quadratic and all the coefficient matrices have been defined and formed in Section 5, the partial derivatives $\frac{\partial h(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}$ and $\frac{\partial h(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}}$ are computed directly:

$$\frac{\partial h(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} = Y_{inequ} + \{\mathbf{u}^{0,T} F_{inequ}^i\} + \{F_{inequ}^i \mathbf{u}^0\}^T + \{F_{inequx}^i \mathbf{x}^0\}^T, \text{ and} \quad (\text{I.18})$$

$$\frac{\partial h(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} = Y_{ineqx} + \{\mathbf{x}^{0,T} F_{ineqx}^i\} + \{F_{ineqx}^i \mathbf{x}^0\}^T + \{\mathbf{u}^{0,T} F_{inequx}^i\}. \quad (\text{I.19})$$

The computation of the derivative $\frac{d\mathbf{x}}{d\mathbf{u}}$ has been illustrated in Appendix H.1:

$$\frac{d\mathbf{x}}{d\mathbf{u}} = -\left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}}\right)^{-1} \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}. \quad (\text{I.20})$$

Back substitution in $\frac{d\mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}}$ yields:

$$\frac{d\mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} = \frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} - \frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}}\right)^{-1} \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}. \quad (\text{I.21})$$

Note that $\frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}}\right)^{-1}$ is the co-state vector that is independent of the control variables, and it is pre-computed at the present operating point. The vector is represented by $\hat{\mathbf{x}}_h$, and we have:

$$\hat{\mathbf{x}}_h = \frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}} \left(\frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{x}}\right)^{-1}. \quad (\text{I.22})$$

Therefore, the linearization of the inequality constraints around the present operating point is:

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) \cong \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0) + \left(\frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} - \hat{\mathbf{x}}_h \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}\right) (\mathbf{u} - \mathbf{u}^0). \quad (\text{I.23})$$

Substitute the current operating point $(\mathbf{x}^0, \mathbf{u}^0)$ into (I.23), we have the final expression of the linearized inequality constraint:

$$\mathbf{a}\Delta\mathbf{u} + \mathbf{d}_h \leq 0 \quad (\text{I.24})$$

where $\Delta\mathbf{u}$ is the increment of the control variable \mathbf{u} ,

$$\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}^0, \quad (\text{I.25})$$

\mathbf{a} is the linear coefficient matrix of $\Delta\mathbf{u}$,

$$\mathbf{a} = \frac{d\mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{d\mathbf{u}} = \frac{\partial \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}} - \hat{\mathbf{x}}_h \frac{\partial g(\mathbf{x}^0, \mathbf{u}^0)}{\partial \mathbf{u}}, \quad (\text{I.26})$$

and \mathbf{d}_h is the constant value vector,

$$\mathbf{d}_h = \mathbf{h}(\mathbf{x}^0, \mathbf{u}^0). \quad (\text{I.27})$$

Distribution

- 2 U.S. Department of Energy
Solar Energy Technologies Office
Attn: Guohui Yuan
M. Kemal Celik
950 L'Enfant Plaza
Washington, DC 20585
- 1 U.S. Department of Energy
Attn: Dan T. Ton
1000 Independence Ave. SW
EE-2A, FORS
Washington, DC 20585
- 1 Electric Power Research Institute
Attn: Brian Seal
942 Corridor Park Blvd.
Knoxville, TN 37932
- 1 National Renewable Energy Laboratory
Attn: Murali Baggu
1617 Cole Blvd.
Golden, CO 80401-3305
- | | | | |
|---|--------|-------------------|------------------------|
| 1 | MS1033 | Jay Johnson | 8812 |
| 1 | MS1084 | Jack Flicker | 5267 |
| 1 | MS1033 | Cliff Hansen | 8812 |
| 1 | MS1027 | Anya Castillo | 5853 |
| 1 | MS1140 | David Schoenwald | 8813 |
| 1 | MS1188 | Mark A. Smith | 8833 |
| 1 | MS0757 | Russell Graves | 6612 |
| 1 | MS0671 | Jordan Henry | 5828 |
| 1 | MS0671 | Trevor Hutchins | 5828 |
| 1 | MS1033 | Abraham Ellis | 8812 |
| 1 | MS1033 | Ross Guttromson | 8812 |
| 1 | MS1104 | Charlie Hanley | 8810 |
| 1 | MS0899 | Technical Library | 9536 (electronic copy) |

