

Reinforcement Learning for Online Adaptation of Model Predictive Controllers: Application to a Selective Catalytic Reduction Unit

Elijah Hedrick^a, Katherine Reynolds^a, Debangsu Bhattacharyya^a, Stephen E. Zitney^b, and Benjamin Omell^c

^a *Department of Chemical and Biomedical Engineering, West Virginia University, 395 Evansdale Drive, Morgantown, WV 26506-6070, USA*

^b *National Energy Technology Laboratory, 3610 Collins Ferry Road, Morgantown, WV 26507, USA*

^c *National Energy Technology Laboratory, 626 Cochran Mill Road, Pittsburgh, PA 15236, USA*

Abstract

This work presents a novel application of reinforcement learning (RL) for online dynamic tuning of model predictive controllers (MPC). Applying a state-action-reward-state-action (SARSA) algorithm for temporal difference learning with a control-specific reward function improves the error tracking performance of a standard MPC formulation. The proposed RL approach is also readily adaptable to other MPCs, or entirely different control approaches. Practical details for the implementation of the RL-MPC algorithm are also presented. The proposed algorithm is applied to a case study of controlling nitrogen oxide (NO_x) emissions in an industrial selective catalytic reduction (SCR) unit, a control problem characterized by significant nonlinearity and time delay. Along with an RL-MPC formulation for NO_x control, another MPC is proposed to mitigate ammonia slip and decrease ammonia consumption in the SCR. Results showing the efficacy of the RL-MPC for NO_x control through learning and implementation on the nonlinear SCR dynamic model are presented.

Key Words

Process Systems Engineering, Reinforcement Learning, Energy, Modeling, Process Control

* Corresponding author.

Tel: 1-304-293-9335, E-mail: Debangsu.Bhattacharyya@mail.wvu.edu

1. Introduction

Research into advanced process control can largely be classified into model-free and model-based approaches. Model-free approaches require only operational data and have been developed to supplement existing control strategies and as standalone implementations (Miccio and Cosenza, 2014). Model-based approaches require a model of the controlled system and can include offline model optimization. However, model-based approaches have generally focused on the application of model predictive control (MPC), which is the most widely used strategy for industrial multivariable control and requires solving an optimization problem online at each control interval (Mayne, 2014; Qin and Badgwell, 2003). Adaptive control implementations where the controller must re-adapt with no memory of past actions have been investigated for both model-free and model-based approaches (Morinelly and Ydstie, 2016; Pan et al., 2007). However, one concept that has not seen such significant coverage in the literature is learning from systems as the controller operates. Another issue in advanced control development is the determination of appropriate tuning parameters, which is not as straightforward as when considering traditional control methods. Further, development of methods for adaptive tuning parameters is not well addressed and is important for flexible operation in systems with large and nonlinear operating spaces (e.g., power production under load-following). Reinforcement Learning (RL) is a data-driven, model-free approach to adaptive control where memory of past control actions can be exploited for further improved control and determination of state-dependent control policies. In this work, RL will be combined with MPC for online, adaptive determination of tuning parameters.

Research on control via RL has seen significant activity in recent years. RL is an adaptive optimal control method whereby an agent learns from its actions on a system. RL is attractive for control of process systems because of its capability to learn and adapt in real-time via only process signals rather than labeled data, as is the case in the training of many other machine-learning methods. However, RL does suffer from sample inefficiency, which is well documented in the literature (Nian et al., 2020; Shin et al., 2019). While significant progress has been made to show the efficacy of RL in learning tasks from no prior knowledge, significant work remains in application of RL to process control (Silver et al., 2017). From the control perspective, two main categories can be observed: those considering RL for direct control and those coupling it with MPC for improved performance.

There are a growing number of examples aimed at direct implementation of RL for process control. Kim et al. propose the use of RL for direct control (Kim and Lee, 2020), where the value function of the RL agent is a Lyapunov function of the controlled system with guaranteed stability. Using this framework and a linear quadratic cost function, an optimal control problem is then formulated using both single-layer and deep neural networks. Spielberg et al. have published work applying deep RL for process control and

provide some commentary on appropriate reward functions for process control applications (Spielberg et al., 2019). Petsagkourakis et al. apply an actor-critic structure for batch bioprocess optimization (Petsagkourakis et al., 2020). An interesting aspect of this work is the use of offline learning to initialize the weights of the function approximators in silico before applying the RL controller to the real plant, an approach that will also be taken in this work. Recently, Yoo et al. have applied a deep deterministic policy-gradient approach to handle system uncertainty in a batch process (Yoo et al., 2021). A two-stage approach where learning is first conducted offline via inverse RL has also been proposed by Mowbray et al., allowing for leverage of process data to reduce the computational load of the online RL agent (Mowbray et al., 2021). Some work has also been done to investigate the online tuning of a traditional process controller, where Brujeni et al. used the SARSA algorithm to tune PI controllers and a parametric, quadratic function approximator is used along with another two-stage approach to implementation (Bruneji et al., 2010).

While direct application of RL to control is desirable, there are still theoretical gaps in providing worst case performance guarantees under exploration, handling of constraints, and in providing assurance of robustness and stability. For these reasons this work focuses on augmenting existing control technology (in the form of MPC) with RL for better performance through a hybrid approach.

Three recent review papers have covered the similarities and opportunities for synergetic application of RL and MPC for process systems (Görges, 2017; Nian et al., 2020; Shin et al., 2019). These works take note of the fact that MPC and RL both seek to achieve optimal performance of a controller in a forward-looking manner, albeit in different ways. It is also of note that, while RL does not require a model, it can be applied with a model or alongside an MPC for improved performance. Most of the work in this area has been in the application of RL for identification or improvement of the controller model of a connected MPC. Morinelly and Ydstie applied RL in this way for dual-MPC of a system with uncertain dynamics, using the RL agent to improve both the MPC model and cost-to-go function at each timestep (Morinelly and Ydstie, 2016). In two related papers, Zanon and Gros et al. have taken a similar approach to the online improvement of the model of an economic (E)MPC (Gros and Zanon, 2020; Zanon et al., 2019). In these works, the MPC is used to give structure to the state-action and value functions of the RL agent, and policy improvement is used to learn better model parameters for the MPC. Kamthe et al. also apply RL to learn a control model for an associated MPC (Kamthe and Deisenroth, 2018). Given that RL can efficiently handle probabilistic systems, this approach learns a transition probability model given a state and an input, with the input to the process plant then calculated via MPC. Shah et al. approximate a model-based control via fuzzy Q-learning for specification of parameters for an underlying proportional-

integral-derivative (PID) controller (Shah and Gopal, 2016). The fuzzy inference system is used to learn an ad hoc model for the PID parameter set, yielding improved control performance.

One critical requirement for the good performance of any MPC is the appropriate choice of tuning parameters. In typical MPC, tuning parameters are the objective function weights and prediction and control horizons. With other modified MPC formulations, there can be additional parameters that are tuned for the desired performance of the MPC, such as model parameters or parameters in the constraint formulation. For many systems, the tuning parameters that can lead to desired results can vary greatly depending on the state of the system, desired trajectory, and specific disturbance rejection. While these tuning parameters are seldom optimized in industrial applications, they are typically tuned manually by the controls engineers to obtain improved performance based on the results from a control task. There can be many discrete sets of tuning parameters found to achieve the desired control performance. There are two main difficulties in this manual tuning approach. First it is difficult to learn and improve control performance by manually adjusting the tuning parameters and observing controller response; the learning outcome can differ significantly based on individual knowledge, skills, or expertise; and many learning opportunities may be missed if tuning is manual. Second, it is difficult to identify when a similar control problem arises in the future. Since the dynamics of a system depend not only on the perturbations but also on the current state of the system, manual identification of similar control problems can be difficult and error prone. In this work, RL algorithms are developed for automatic learning and retention of MPC tuning parameters based on the control performance as well as use of those tuning parameters as initial values when a similar control problem arises via a mapping of the tuning parameters onto the output error. This has the added effect of determining an ad hoc scheduling of the parameters with respect to the error, allowing for selection of the best parameters at each time step rather than having them remain static.

Because of the explorative requirements of the learning process for RL, a two-stage algorithm is developed to avoid unacceptable performance when controlling under an RL agent without sufficient knowledge of the system. In addition, the shift from offline episodic learning to online continuous learning is made when the controller is applied to the true plant.

The performance of the RL-MPC is studied via application to a selective catalytic reforming (SCR) unit that reduces nitrogen oxides (NO_x) in the exhaust gases from a coal-fired power plant by reacting with injected ammonia (NH_3). The SCR unit poses unique control challenges due to nonlinearity of the reaction mechanism and adsorption-desorption dynamics of NH_3 at the solid catalyst surface. While the primary control objective of the SCR control is to satisfy NO_x emissions constraints by manipulating the injection of ammonia at the reactor inlet, it is also desired to minimize the NH_3 slip (i.e. unreacted NH_3 at the SCR outlet) to reduce the operating costs and to meet environmental regulations (*Estimating*

Ammonia Emissions from Stationary Power Plants, 2009). A feedback-augmented feedforward (FBAFF) control strategy using a PID controller for feedback control and a simple transfer function type model for computing the feedforward contribution is a common control strategy for SCR control in power plants (Leopold, 2010). This strategy does not explicitly take ammonia slip into account. In two papers, Zhang et al. have presented an MPC formulation with gain scheduling for handling of the nonlinearity of the SCR. In the first paper, the SCR is simulated as a continuously stirred tank reactor and the control model is developed from input-output data on this simplified case, though in reality the SCR will follow the operating characteristics of a plug-flow or packed-bed reactor given its configuration (Zhang et al., 2016). The second paper shows the in-situ implementation of the MPC on an SCR in a power plant and presents a series of operating scenarios (Zhang et al., 2018). The paper includes results for comparison of the MPC to an FBAFF strategy and shows the ability of the MPC to mitigate constraint violation. Reduction of ammonia consumption of up to 25% is also observed. Shah et al. present linear and nonlinear MPC studies of the startup of an SCR (Shah et al., 2015). Their results are not necessarily comparable to the work in this study given that they studied startup rather than deviation from a nominal operating point. Ammonia slip is accounted for via a bound constraint in both the linear and nonlinear MPC formulations. Wu et al. use linear MPC with subspace identification to control an SCR under unknown disturbances, however ammonia slip is not considered and the details of the SCR plant model are not provided (Shen et al., 2012). Finally, Peng et al. implement a nonlinear MPC via system identification from industrial SCR data (Peng et al., 2006). They identify a sufficient model for control of the SCR from the available data and conduct control studies on a version of the controller model with modified parameters. Because the parameters are changed in a simplistic way, this may still not reflect the actual dynamics of the plant and may not expose any model mismatch.

In the work presented here, the RL algorithm is applied to an MPC with a conventional control objective applied to the SCR. This MPC the conventional setpoint control for the SCR outlet NO_x concentration. Another MPC control objective considers minimization of consumption of ammonia in the SCR while considering the outlet NO_x concentration as a constraint. The remaining sections of this paper are arranged as follows. In Section 2, the RL-MPC algorithms are described. In Section 3, the SCR model is discussed. In Section 4, results are presented followed by conclusions in Section 5.

2. RL-MPC Algorithm

In this section, first the basics of RL are discussed followed by the proposed RL-MPC structure and algorithm. Tuning and implementation of the RL-MPC algorithm are then discussed.

2.1 RL Basics

RL is a machine learning (ML) method that differs from other ML methods in that, rather than learning from guided actions or labeled data, the agent learns directly from taking actions on the system as shown in Figure 1.

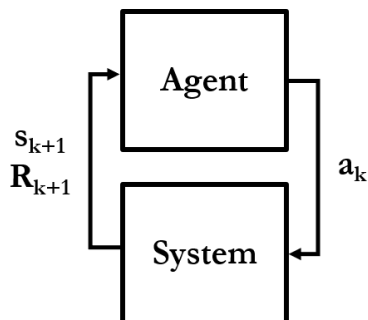


Figure 1. RL Diagram

The key elements of RL are the *policy* (π), the *reward signal* (R), and the *value function* (V or Q), each important to selecting actions given a state of the system and subsequently updating what the agent learns about the system (Sutton and Barto, 2018). The *policy* defines what action is selected by the agent given the state of the system. The *reward* is a scalar quantification of the quality of the action that was taken given the perceived change in the state of the system under that action. The *value function* defines the value of a given state or, in cases such as those considered in this work, the quality of state-action pairings. Conventionally, the state-action value function is referred to by $Q(s,a)$ – or $\hat{q}(s,a)$ when combined with function approximation – where s is the state and a is the action. This convention is followed in this work, where the function is referred to interchangeably as either the state-action value function or the quality function. The state-action value function is formally defined as the expected sum of rewards, discounted by the parameter γ , taking action a in state s and following the policy π from that point, as in Equation 1.

$$Q^\pi(s_t, a_t) = E^\pi \left\{ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid s_t = s, a_t = a \right\} \quad (1)$$

In RL approaches such as those used in this paper, the goal of the RL agent is to learn a value function to maximize the discounted cumulative reward (G) under the current policy (Sutton and Barto, 2018).

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (2)$$

It is also important to note that RL can be model-free or can include a model of the system. While the algorithm presented in the following sections details an RL agent working cooperatively with a model-based controller, the agent itself is assumed to have no access to the controller model and is hence a model-free implementation. RL methods can be tabular or approximate. Tabular methods (considering state-action pairs) compute and update an explicit value for each pairing. These methods can be powerful in cases where the state and action spaces are discrete, finite, and do not necessarily grow with time. However, tabular methods suffer greatly as the state and action spaces grow, or if they become continuous. Approximate methods, conversely, use function approximation to give value to the state-action value function, yielding approximations that may span many states (i.e., having reduced accuracy) but allowing for consideration of larger systems. Since the primary focus of this work is on continuous process systems, an approximate RL method is used.

2.2 RL-MPC

The application of RL to MPC presented here focuses on the online tuning of an underlying MPC with the RL agent. Figure 2 shows the control diagram for the proposed structure, where it is important to note that the **RL-Agent** block is the same as the agent in Figure 1, the combined **MPC** and **Plant** are the same as the system in Figure 1, and the underlying MPC structure is any MPC controlling a plant. To avoid confusion, the word ‘*system*’ is used from here to refer to the combined MPC and plant and is the subject of the RL agent, while the word ‘*plant*’ is used to describe the physical representation of the actual plant controlled by the MPC. In all cases, the objective of the RL agent is to find, given any system state, the ‘best’ MPC horizons (control and prediction, following the normal definitions).

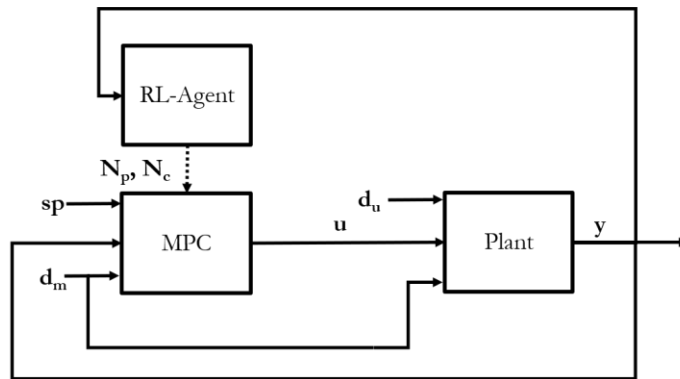


Figure 2. RL-MPC Control Diagram

Here, sp is the reference signal, d_m and d_u are the measured and unmeasured disturbances, N_p and N_c are the prediction and control horizons, u is the vector of inputs, and y is the vector of outputs.

The controller horizons have been selected as the active tuning parameters for the RL agent to control because they are those that would be most typically exposed for user tuning in an existing MPC. Other options could include model parameters or objective function weights. To use RL as a method for learning and adapting the model parameters requires satisfying aspects such as richness of data and properties such as sufficient smoothness of the model and stability of the resulting. The objective function weights are also not addressed because weights such as those applied to the control moves may need to take into account equipment wear and tear due to excessive control moves and thus may require user validation for acceptable values. Furthermore, those variables are continuous valued, and would result in a more challenging problem for online application when compared to a discrete and bounded set of parameters.

2.3 Algorithm

The proposed RL-MPC algorithm is an application of the approximate State-Action-Reward-State-Action (SARSA) algorithm for temporal difference learning (Sutton and Barto, 2018). The proposed scheme in this work involves two stages- an offline learning stage used for initialization followed by an online stage. The algorithm starts with offline learning because of the desired high exploration by RL for learning. In the offline learning, even though ϵ in the ϵ -Greedy search decays exponentially, its initial value is high, which is acceptable because no performance bounds are considered during offline learning. If the controller performs very poorly as a result of such exploration, it only adds to what the agent knows about those actions in those states via the update on the estimate of \hat{q} . In the online case, such a high level of exploration is not allowed (i.e., ϵ is constrained to a small value), since significant performance loss due to exploration is unacceptable to an operating plant.

The structure of the SARSA-based RL-MPC algorithm can be seen in Algorithm 1, where offline episodic learning is considered. The inputs to the algorithm include the value function approximator, RL hyperparameter values, and the MPC to control the plant. The basis functions used for the approximation of the state-action value function are Gaussian radial basis functions, as seen in Equation 3.

$$q_i(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|c_i-x\|^2}{2\sigma^2}} \quad (3)$$

where σ is the shape parameter and c_i is the i^{th} basis function centroid. These basis functions are combined linearly to find the quality function value as shown in Equation 4.

$$\hat{q}(s, a) = w^T q = \sum_{i=1}^d w_i q_i \quad (4)$$

The reward function considered for all cases in this work is the negative of the squared error of the desired plant output (setpoint) and the actual plant output at any point in time, as seen in Equation 5.

$$R_k \equiv -\|y_{sp} - y_{k+1}\|^2 \quad (5)$$

The squared error at each timestep was selected because it is always positive (yielding an always-negative reward signal) and because the goal of the algorithm in this work was only to improve performance with respect to output error. More generally, other reward signals could be designed to reward desirable control moves, though it should be noted that the reward is taken at a single time instance rather than over the entire trajectory.

Algorithm 1. Algorithm for RL-MPC Offline Learning

RL-MPC Offline Algorithm

Inputs:

$\hat{q}(s, a)$ a value function approximator with weights \mathbf{w}

$\alpha \in (0, 1]$, $\varepsilon \in [0, 1]$, $\gamma \in [0, 1]$

Arbitrarily initialize \mathbf{w} , the weights of the function approximation

An MPC to control the plant

For each episode:

Initialize the plant and controller to steady-state (i.e. $y = 0$, $u = 0$ in deviation variables)

Selection an initial action (a_0) under the current policy (i.e. ε -Greedy)

For each timestep (k) of each episode:

1. Apply $a_k = \begin{bmatrix} N_p \\ N_c \end{bmatrix}_k$ to the MPC
 2. Calculate:

$$u_k = u^* \text{ from the MPC}$$
 3. Apply u_k to the plant and get y_{k+1}
 4. Calculate $R_k = R(y_{k+1})$
 5. Select a_{k+1} under the current policy (i.e. ε -Greedy)
 6. Calculate $\delta \leftarrow R_k + \gamma \hat{q}(y_{k+1}, a_{k+1}) - \hat{q}(y_k, a_k)$
 7. Update $\mathbf{w} \leftarrow \mathbf{w} + \alpha \delta \nabla \hat{q}(y_k, a_k)$
-

For each episode of the training, the plant and controller are initialized to steady-state conditions and an initial action vector is selected. The action vector (a_k) selected by the RL agent given the state of the plant includes the prediction horizon (N_p) and control horizon (N_c) parameters for the MPC. At each time step of a training episode, an action is applied to the MPC and the MPC calculates the states (u_k), which are then applied to the plant to get the new outputs (y_{k+1}). The policy used to select the actions is the ϵ -Greedy policy, seen in Algorithm 2 (Sutton and Barto, 2018). Because persistently changing disturbances (d_m, d_u) are considered during the episodic training of the agent in this work, no terminal state is considered, and each episode exits when the maximum number of timesteps is reached. For any task where a logical terminal state exists (e.g. a step change in the setpoint), the termination of an episode would simply include this option.

Algorithm 2. ϵ -Greedy Search Algorithm

ϵ -Greedy Search Algorithm

Inputs:

$\epsilon \in (0,1]$

For each evaluation:

1. Draw P from a uniform distribution
2. If $P > \epsilon$

$$a_k = \operatorname{argmax}_a \hat{q}(s, a)$$

If $P < \epsilon$

select $a_k = a \in A$ randomly

Break ties randomly

2.4 On configuring the RL agent

The structure and tuning of the RL algorithm are important when considering the numerical properties of the learning that is taking place. It is desired that all values used in the approximation of the quality function (i.e. the state and action values, along with the basis function centers and the reward function), be scaled, normally on $[-1, 1]$ or $[0,1]$. This scaling ensures a consistent numerical basis for function approximation, the reward signal, the value function, and the gradient of the value function. Under this assumption, along with the assumption that the basis function centers are evenly distributed in each dimension, the shape parameter of the Gaussian distribution (σ) is commonly taken to be given by Equation 6, where n is the number of discretizations in a single direction (Konidaris et al., 2011).

$$\sigma^2 = \frac{2}{n-1} \quad (6)$$

The discount rate parameter in Algorithm 1, $\gamma, \in [0,1]$ reflects the user preference, where the lower limit favors only immediate rewards, with increasing value favoring future rewards more as exhibited in the calculation of the temporal difference, δ , as seen in Table 1 (Sutton and Barto, 2018).

Tradeoff between exploration and exploitation is achieved by the use of the ε -Greedy algorithm for action selection, as shown in Algorithm 2. In general, the value of ε can simply be selected to be small, allowing for some exploration but most of the time taking the greedily-selected action. However, when the knowledge about the system is poor, like at the beginning of initiating the agent, exploration is more desired while exploitation of the learning becomes more dominant as time goes on, as seen in Equation 7:

$$\varepsilon = \varepsilon_0 + \varepsilon_1 e^{-t} \quad (7)$$

where ε_0 is small and can be taken to be the final desired rate of exploration, and ε_1 is used to drive early exploration (Carlucho et al., 2017).

One of the important considerations in tuning the RL hyperparameters is that of the step size, α in Algorithm 1. If the step size is poorly selected, learning can fail to proceed satisfactorily or can even diverge. Conventionally, the step size is selected to correlate to the desired rate of learning or, more practically, the approximate number of episodes targeted for learning (Sutton and Barto, 2018). This is, however, only an approximation and learning results may vary greatly in practice. Another option is to allow the step size parameter to decay with the number of elapsed episodes, corresponding with the desired progress of learning. In early episodes it is desirable to explore significantly and take larger steps in parameter value; then, as more interaction takes place, more actions are selected greedily and smaller step sizes refine what has been learned. This is the option selected for this work and is shown in Equation 8:

$$\alpha = \alpha_0 \frac{1}{N_{ep,elapsed}} \quad (8)$$

2.5 On implementation

Several aspects need to be taken into account on implementing the RL-MPC algorithm. First, the closed-loop system with RL-MPC implemented should be stable. In this work, it is assumed that the learning is constrained to the admissible sets of parameters (i.e., horizons) to guarantee stability of the resulting closed-loop system. Second, while the RL agent can be applied to an MPC acting on any plant,

initialization of the agent's actions can be important, especially for systems where control performance cannot degrade below some limit due to safety, quality, or other concerns, even under initial conditions. For the RL-MPC in this study, a two-stage approach is proposed for initializing the RL agent. In the first stage, the reduced model identified for use in the underlying MPC is leveraged. Using this model as the plant, episodic learning is performed offline. At the second stage, the RL-MPC is implemented on the real plant using the following two methods applied in this study. In one method, the policy learned offline is called stationary and online actions are always selected greedily with no additional learning. In the second method, learning continues online in the real plant setting, albeit with lesser exploration than allowed earlier during offline episodic learning.

After the offline learning has completed, the RL agent must now learn online from control of the plant. While the same general RL structure can be used and the offline learning is retained, real operation of the plant does not necessarily include episodes as defined by a fixed starting state and terminal condition. Because of this, it is necessary to slightly alter the RL algorithm to account for its application to an operating plant. In the online continuation of the RL-MPC algorithm shown in Algorithm 3, the temporal difference, δ , is updated based on deviation from the average reward, \bar{R} (step 6). At each sampling instant, the average reward is updated based on another step size parameter, β (step 7). The weight update in this setting remains unchanged except for the removal of discounting which will have no impact in online learning (step 8). The structure of the action-value function approximation also remains unchanged. Selecting actions amounts to maximizing the action-value function over the actions in any state.

Algorithm 3. RL-MPC Online Control and Learning Algorithm

Online RL-MPC Algorithm

Inputs:

$\hat{q}(s,a)$ that is differentiable, consistent with offline learning

$\alpha \in (0,1)$, $\varepsilon \in [0,1]$, $\beta \in (0,1)$

\mathbf{w} from offline learning, $\bar{R} = 0$

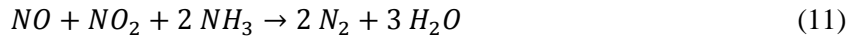
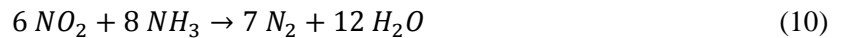
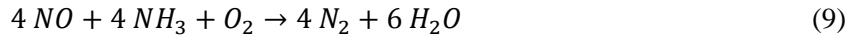
An MPC to control the plant, consistent with offline learning

For each sampling time (k), continuing while online:

1. Apply $\mathbf{a}_k = \begin{bmatrix} N_p \\ N_c \end{bmatrix}_k$ to the MPC
 2. Calculate:
 $\mathbf{u}_k = \mathbf{u}^*$ from the MPC
 3. Apply \mathbf{u}_k to the plant and get \mathbf{y}_{k+1}
 4. Calculate $R_k = R(\mathbf{y}_{k+1})$
 5. Select \mathbf{a}_{k+1} under the current policy (i.e. ε -Greedy)
 6. Calculate $\delta \leftarrow R - \bar{R} + \hat{q}(\mathbf{y}_{k+1}, \mathbf{a}_{k+1}, \mathbf{w}) - \hat{q}(\mathbf{y}_k, \mathbf{a}_k, \mathbf{w})$
 7. Update $\bar{R} \leftarrow \bar{R} + \beta\delta$
 8. Update $\mathbf{w} \leftarrow \mathbf{w} + \alpha\delta\nabla\hat{q}(\mathbf{y}_k, \mathbf{a}_k, \mathbf{w})$
-

3. Case Study: Industrial SCR Unit

Selective catalytic reduction (SCR) is a common technique for the treatment of flue gas produced by the combustion of fossil fuels in thermal power plants to reduce emissions of NO_x , namely nitric oxide (NO) and nitrogen dioxide (NO_2) (Nova et al., 2005). SCR involves the reduction of NO_x into N_2 and H_2O via reactions with NH_3 in the presence of O_2 , commonly on a vanadium-tungsten-titania catalyst. These reactions take the following forms:



Depending on the operating conditions of the SCR unit, additional side reactions can occur, most of which involve the oxidation of NH_3 to various compounds, including the primary oxidation reaction where NH_3 is reduced to molecular nitrogen (N_2) and water vapor (H_2O).



It is generally accepted that these reactions follow an Eley-Rideal mechanism at the conditions found in SCR units operating in thermal power plants, with the resulting adsorption-desorption dynamics leading to significant time delay through the SCR reactor in many cases (Lietti et al., 1997). Additionally, SCR can be limited by both external and internal mass transfer, yielding an overall highly nonlinear system that exhibits significant time delay (on the order of 15 minutes).

For evaluating the performance of the proposed RL-MPC, a detailed first-principles model of the SCR unit is developed that can well represent the nonlinearities and control challenges of this system. Within the open literature, many first-principles SCR unit models exist that focus on steady-state operation (Beekman and Hegedus, 1991; Beretta et al., 1998). A more recent development in this area by Shen et al. includes detailed kinetic rate expressions for the main SCR reactions, though it does neglect the NH_3 oxidation reaction, and examines the influence of catalyst structure on the efficiency of SCR reactors (Shen et al., 2012). Kanniche et al. developed a more sophisticated model that includes not only NH_3 oxidation but sulfur dioxide oxidation, which can be of more concern if the SCR unit is not maintained at a high enough temperature (Kanniche et al., 2011). The model further considers internal and external diffusion within and surrounding the catalyst bed. A significant contribution of this model is that it is validated with industrial data at the reactor outlet for a few different cases, making it one of very few examples of industrial validation of a first-principles SCR unit model available in the open literature. Steady-state analysis of the SCR unit, while sufficient to study base-load conditions where the dynamics are generally small in magnitude, is not able to capture the behavior of the SCR unit under power plant load-following conditions.

The dynamics of the SCR reactions were investigated by Lietti et al. where the results indicated that the transient kinetics were best described by an Eley-Rideal mechanism (Lietti et al., 1997). These kinetic expressions account for the adsorption-desorption of NH_3 on the catalyst, which yields the time-delay responses inherent to the SCR system, however they do not account for the oxidation of NH_3 . Subsequent work by Nova et al. expands on the work of Lietti et al. to incorporate this side reaction, though neither work includes the effects of internal or external diffusion limitations or is validated with industrial data (Nova et al., 2000). Qin et al. later used these same SCR reaction rate expressions in their model, which was validated with some industrial data from a 1000 MW ultra-supercritical coal power plant (Qin et al., 2016). However, since the reaction rates used were from Lietti et al., the mass transfer limitations were not considered, which could account for some of the discrepancies between their model predictions and their validation data. Muñoz et al. sought to address this issue with their model that considers internal and external mass transfer alongside the SCR reactions to model both lab- and bench-scale SCR reactors

(Muñoz et al., 2015). This model neglects NH_3 oxidation and is only validated with experimental data from lab- and bench-scale systems, though an industrial catalyst is used within those systems. The lack of validation with industrial data is an overarching gap in the literature for first-principles SCR models.

A newer approach to modeling the SCR unit dynamics is to use machine learning techniques such as neural networks or support vector machines to avoid the complexity of first-principles models, especially for use with controls where such complex models may be unable to solve quickly enough for online execution (Lv et al., 2013; Safdarnejad et al., 2019). However, most data-driven models of this type tend to suffer from their inability to extrapolate during new operating regimes. Lv et al. attempted to address this problem by adding the capability to update their model when encountering unrecognized operating regimes (Lv et al., 2020). However, performance of the model outside of the original operating space is then subject to the respective performance of the correction; this may lead to unacceptable losses in predictive capabilities, especially far from the original operating point. This represents a distinct disadvantage of using such models, as the SCR unit dynamics are complex and dependent on the path of operational changes to a large extent. Thus, as more thermal power plants start utilizing a load-following strategy, the amount of time spent in previously unencountered operating points will increase as will the need to predict the behavior of the SCR unit under these conditions.

The key difference in the first-principles SCR dynamic model considered here is that it accounts for the mass transfer limitations in addition to not only the main SCR reactions but also the undesired NH_3 oxidation reaction, all of which contribute to the dynamic behavior of the unit. Furthermore, the model is validated with the industrial data.

In this work, two MPC formulations are presented for the control of the SCR unit. Similar to much of the work above, the first MPC is a setpoint tracking MPC in which the objective of the controller is simply to reject disturbances in the outlet NO_x concentration. However, this implementation also takes into account modeling for some of the common (and measurable) disturbances seen in SCR units for more effective load-following control. This controller is also used to show the efficacy of implementing reinforcement learning (discussed in the next section) in conjunction with MPC. The second MPC developed here minimizes the ammonia slip while treating the NO_x at the outlet of the SCR as a constraint, thereby handling the non-square control problem of accounting for both NO_x and ammonia emissions.

This section of the paper will detail the development of the SCR plant model, followed by a discussion of the industrial standard in SCR control along with development of the two MPCs proposed for control of the unit.

2.1 SCR Dynamic Modeling

The dynamic SCR model developed herein considers both the kinetic and mass transfer limitations of the SCR reactions over a catalyst and is suitable for modeling the operation of SCR units in supercritical pulverized coal (SCPC) power plants. Developed in Aspen Custom Modeler (ACM), the one-dimensional distributed-parameter reactor model is discretized along the length of the SCR unit and uses the Peng-Robinson equation of state to calculate the physical properties of the gas phase. The catalyst is an industrial vanadium-tungsten-titania which is typically suitable for use in a temperature range of 280 – 400°C (Nova et al., 2000).

Due to the relatively small (parts-per-million) concentration of NO_x in the flue gas, the reactor is also assumed to be isothermal. The catalyst bed is assumed to be made of monolithic channels, across which the pressure drop is usually small. In SCPC power plants, the ratio of NO to NO₂ produced is sufficiently high enough, especially with recent advancements in low-NO_x burners and other technologies to reduce emissions, to consider all NO_x produced as NO alone (Mitchell, 1998). Thus, the SCR main reactions are considered as the reduction of only NO with NH₃, resulting in the modelling of only the NO reduction and NH₃ oxidation, Eqns. (11) and (14), respectively. These key assumptions are summarized as follows:

1. Axial dispersion is neglected
2. Isothermal
3. Pressure drop is neglected
4. All NO_x considered as NO
5. Only NH₃ is adsorbed on the catalyst

The kinetic expressions for the rates of NO reduction, NH₃ oxidation, and NH₃ adsorption and desorption are adapted from Nova et al. (Nova et al., 2000) and Lietti et al. (Lietti et al., 1997), though since the model here considers external mass transfer, the surface concentrations of NH₃ and NO are used in the rate expressions rather than the bulk concentrations.

$$r_{red} = k_{red} \exp\left(\frac{-E_{red}}{RT}\right) C_{NO,s} \theta_{NH_3}^* \left(1 - \exp\left(\frac{-\theta_{NH_3}}{\theta_{NH_3}^*}\right)\right) \quad (13)$$

$$r_{ox} = k_{ox} \exp\left(\frac{-E_{ox}}{RT}\right) \theta_{NH_3} \quad (14)$$

$$r_{ads} = k_{ads} C_{NH_3,s} (1 - \theta_{NH_3}) \quad (15)$$

$$r_{des} = k_{des} \exp\left(\frac{-E_{des}}{RT}\right) \theta_{NH_3} \quad (16)$$

Within these kinetic expressions θ_{NH_3} is the surface coverage of ammonia and $\theta_{NH_3}^*$ is the critical surface coverage, above which the coverage of ammonia no longer affects the rate of the reduction reaction. The activation energy for the desorption of NH_3 (E_{des}) is considered such that it has a dependence on θ_{NH_3} .

$$E_{des} = E_{des}^0(1 - \alpha\theta_{NH_3}) \quad (17)$$

Since the surface coverage of NH_3 governs the behavior of the reaction kinetics, an unsteady balance is also carried out for the surface coverage of NH_3 to account for its dynamic behavior, which is essential for capturing the overall dynamics of the SCR unit. It should be noted that due to the consideration of the internal diffusion, the internal effectiveness factor (η_{int}) is accounted for here.

$$\frac{\partial\theta_{NH_3}}{\partial t} = (r_{ads} - r_{des} - r_{red} - r_{ox})\eta_{int} \quad (18)$$

For more details on the mass transfer modeling, see Appendix A.

Fundamental gas phase unsteady mass balances are written as follows:

$$\frac{\partial C_{NO}}{\partial t} = -v \frac{\partial C_{NO}}{\partial z} + \frac{a}{\varepsilon_b} k_{g,NO} (C_{NO,s} - C_{NO}) \quad (19)$$

$$\frac{\partial C_{NH_3}}{\partial t} = -v \frac{\partial C_{NH_3}}{\partial z} + \frac{a}{\varepsilon_b} k_{g,NH_3} (C_{NH_3,s} - C_{NH_3}) \quad (20)$$

$$\frac{\partial C_{N_2}}{\partial t} = -v \frac{\partial C_{N_2}}{\partial z} + \Omega_B \eta_{int} (r_{NO} + 0.5r_{ox}) \left(\frac{1 - \varepsilon_b}{\varepsilon_b} \right) \quad (21)$$

This entire partial differential-algebraic equation system is solved using the method-of-lines approach in ACM to determine the one-dimensional, dynamic profiles in the SCR unit.

Since the kinetic rate expressions and mass transfer models adapted for use here have only been validated against lab- or bench-scale results, the model was validated against industrial operating data prior to its use for control studies. The parameters listed in Table 1 were used for this validation, which utilizes available data from the open literature on catalyst parameters for an industrial vanadium-tungsten-titania catalyst used in SCR high dust applications, such as SCPC power plants (“Coal - DNX HD - SCR DeNOx catalyst DNX®-series,” 2020). In addition, parameters from Lietti et al., Nova et al., and Muñoz et al. were used for the reaction rate expressions and mass transfer models (Lietti et al., 1997; Muñoz et al., 2015; Nova et al., 2000).

Table 1. SCR Model Parameters

Parameter	Value	Unit
$D_{H,cat}$	9	mm
E_{des}°	23	kcal / mol
E_{ox}	28.8	kcal / mol
E_{red}	19.2	kcal / mol
k_{ads}	33.87	$m^3 / mol \cdot s$
k_{des}	$2.20 \cdot 10^6$	s^{-1}
k_{ox}	$3.25 \cdot 10^6$	s^{-1}
k_{red}	$1.18 \cdot 10^8$	$m^3 / mol \cdot s$
s_{cat}	11	mm
α	0.405	–
ε_b	0.82	–
$\theta_{NH_3}^*$	0.06	–
τ	5	–
Ω_{NH_3}	270	mol_{NH_3} / m^3

The industrial data against which the SCR model is compared was first filtered through a low-pass Butterworth filter, which reduced the noise associated with the measurements. Then, a moving average filter was applied to smooth out any large variations in the measurements (i.e. variations larger than noise but not relevant to the overall trend of the data), which further reduced the noise and aided in the convergence of the model. Once the data was treated, the model was simulated with the filtered inputs from the industrial data and the primary output of the model, the outlet NO_x concentration, was compared.

The model validation with industrial data resulted in a normalized root mean square error of 0.251. As can be seen in Figure 3, the NO_x output matches sufficiently with the industrial data, and the SCR model captures well all of the dynamic trends from the data set. Note that the values in this figure have been normalized to protect the proprietary data.

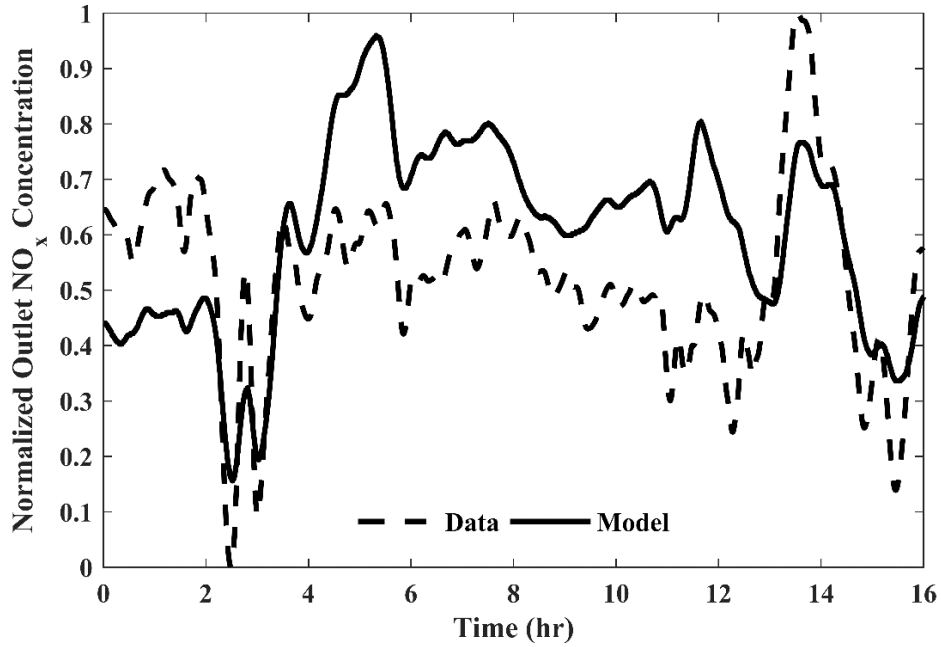


Figure 3. SCR model error, validation study

Thus, for studying the dynamics of the SCR from a control perspective, the dynamic model is considered sufficiently validated.

2.2 Current Industrial SCR Control

In order to evaluate the performance of the controllers developed in this work, the industrial standard control strategy for SCRs was implemented on the same SCR model considered for MPC development. This standard amounts to feedback augmented feedforward (FBAFF) control with varying degrees of complexity in the feedforward model (Leopold, 2010; Ogunnaike and Ray, 1994). For the purposes of this work, the feedforward model is taken to be a constant ratio (taken at the full load value) of the ammonia injection rate to the inlet NO_x flow, as in Equations 22 and 23. This consideration is likely simpler than the model in an industrial control system but accounts for most of the feedforward signal.

$$r_{FF} = \left. \frac{F_{\text{NH}_3}}{F_{\text{NO}_x}} \right|_{\text{Full Load}} \quad (22)$$

$$F_{\text{NH}_3,FF,k} = r_{FF} F_{\text{NO}_x,k} \quad (23)$$

The feedback part of the control is taken to be a well-tuned PID controller. This control structure is implemented entirely using ACM in-built controller blocks.

2.3 MPC of the SCR Unit

Two different MPCs are developed for control of the SCR unit in this work. For both MPCs, data from the SCR model described above is collected for system identification. From the data collected, models for both controllers are identified according to the least squares estimate assuming linear state-space models with C as an identity matrix, as in Equations 24-26 (Bhattacharyya and Rengaswamy, 2010).

$$x_{k+1} = Ax_k + Bu_k \quad (24)$$

$$y_k = Cx_k, C = I \quad (25)$$

$$\Phi = (\Psi^T \Psi)^{-1} \Psi^T Y \quad (26)$$

Table 2 lists the inputs (manipulated variables), outputs (control variables), and disturbances considered in this work.

Table 2. Summary of MPC Variables

Inputs (MVs)	
Inlet Ammonia Flow	u_1
Outputs (CVs)	
Outlet NO _x Concentration	y_1
Outlet Ammonia Concentration	y_2
Modeled Disturbances	
Inlet Flue Gas Flow	d_1
Inlet NO _x Concentration	d_2
Unmodeled Disturbances	
Inlet Ammonia Temperature	d_3
Inlet Flue Gas Temperature	d_4
Inlet Dilution Air Flow	d_5

3.1.1 MPC-1

The first MPC is called MPC-1 and the corresponding RL case is called RL-MPC-1. MPC-1 is a standard linear formulation of a setpoint tracking MPC, as seen in Equation 27 (Qin and Badgwell, 2003).

$$\begin{aligned}
\min_u J &= y_{N_p}^T H y_{N_p} + \sum_{k=0}^{N_p-1} y_k^T Q y_k + \sum_{k=0}^{N_c} u_k^T D u_k \\
s. t. \quad x_{k+1} &= A x_k + B u_k \\
y_k &= C x_k \\
u_{i,lb} &\leq u_{i,k} \leq u_{i,ub}
\end{aligned} \tag{27}$$

where N_p and N_c are the prediction and control horizons, respectively. In this formulation, the goal is to reject disturbances and return the outlet NO_x concentration to its nominal value, thereby keeping NO_x emissions within their environmental constraints. In this formulation, ammonia slip can be reduced but is not explicitly taken into account in the MPC formulation.

3.1.2 MPC-2

The second MPC, called MPC-2, is designed to control the outlet NO_x concentration within a range around the nominal condition while minimizing the outlet NH_3 concentration over the prediction horizon, as seen in Equation 28.

$$\begin{aligned}
\min_u J &= \sum_{k=0}^{N_p} (y_{2,k} - \omega)^T Q (y_{2,k} - \omega) + M p_k^2 + \sum_{k=0}^{N_c} \Delta u_k^T D \Delta u_k \\
s. t. \quad x_{k+1} &= A x_k + B u_k \\
y_k &= C x_k \\
p_k &\geq y_{1,k} - \text{NO}_{x,limit} \\
p_k &\geq 0 \\
u_{i,lb} &\leq u_{i,k} \leq u_{i,ub}
\end{aligned} \tag{28}$$

It should be noted that this is a non-square problem (one input to control two outputs) and explicit consideration of ammonia slip and outlet NO_x concentration is not feasible in typical PID or FBAFF configurations. Because maintaining some excess ammonia is necessary, a minimum desired outlet NH_3 concentration (ω) is included in the objective function to avoid its inclusion as a constraint on the output.

4. Results and Discussion

Open-Loop Responses

Using the validated SCR dynamic model, the open-loop responses of the outlet NO_x and NH_3 concentrations to $\pm 10\%$ step changes in injected ammonia (u_1) and flue gas temperature (d_4) can be seen in Figure 4 (a,b). Open loop responses for the remaining disturbance variables, as well as characterization of gains and time constants for each variable, can be found in Appendix B.

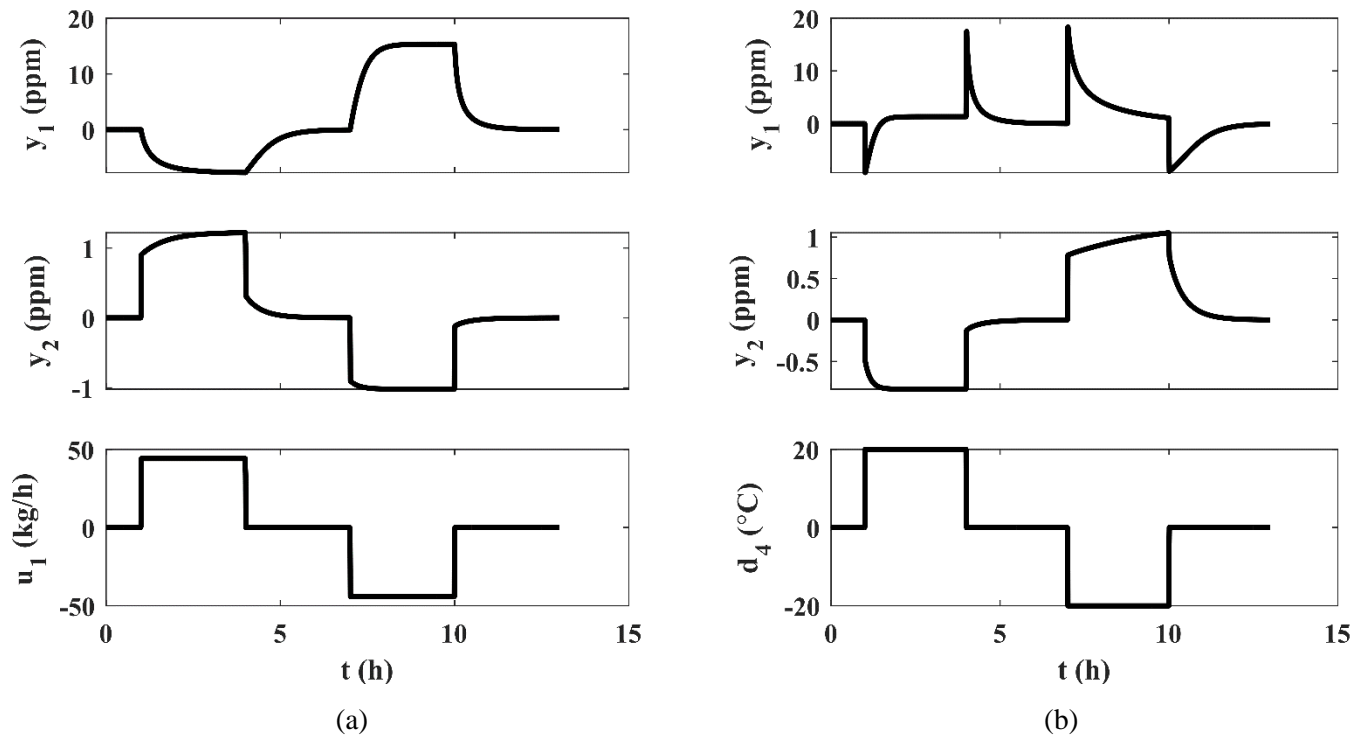


Figure 4. (a) Output Responses to step changes in u_1 , (b) Output Responses to step changes in d_4

As can be seen from the open-loop responses, the dynamics are relatively slow with the SCR plant requiring approximately three hours to return to a steady-state condition after each step change. From the open loop response to the inlet ammonia flow (u_1), the outlet NO_x concentration (y_1) is much more sensitive to the input than the outlet ammonia concentration (y_2). This is also true when observing the response of the system with respect to changes in the flue gas temperature (d_4). In both cases the change in y_1 is approximately one order of magnitude higher. The nonlinearity of the system is also clear when observing the dynamic responses of the outputs to both u_1 and d_4 , where the process gains and response times vary significantly for different steps changes from the nominal operating condition. The time constants, time delays, and gains for the input and each disturbance can also be found in Appendix B.

To study the performance of the proposed controllers, case studies were conducted by considering changes at the inlet boundary conditions as would be expected under load-following operation by the power plant.

Study 1: MPC-1

Learning Results

The first step in the two-phase initialization of the learning agent is to allow the agent to learn offline. To do so, the agent was applied to MPC-1 as described above, and the combined RL-MPC was used for control considering the MPC's control model as the plant. Details on the setup for the RL agent can be found in Appendix C.

To allow the agent to learn, each episode introduced both unmodeled disturbances and modeled disturbances; in both cases, the sign and magnitude of the disturbance were selected at random, though the magnitude was bounded. In all cases, regardless of the presence of modeled disturbances, the only signals available to the RL agent were the output NO_x concentration feedback and setpoint tracking-based reward (Equation 3) at each timestep. In each episode, the initial state was selected randomly on the interval $[-1, 1]$. After sufficient time for the agent to return the plant to steady-state (determined from other studies conducted a priori), two disturbances, separated again by sufficient time to return to steady-state (with this time again determined a priori), were injected into the plant. The disturbances were of random sign and magnitude to promote learning across the entire output space. Subject to the setup and hyperparameters described above, the RL agent was then allowed to operate and learn over the duration of each episode. Each episode was terminated after reaching the specified maximum number of timesteps. A total of 100 episodes was simulated per trial. The total number of episodes and maximum number of timesteps were determined empirically after observing the typical performance of the controller under learning.

For the purposes of evaluating the learning of the RL agent, the discounted cumulative reward (G , as defined previously) is considered, as seen in Equation 29. This is a commonly-used metric for the progress of learning across episodes. Because of the structure of the reward function used here, the discounted reward is expected to start with a large negative value and approach a value close to zero as learning progresses.

$$G_{episode} = \sum_{k=0}^T \gamma^k R_{k+1} \quad (29)$$

Figure 5 shows the results of one learning trial in terms of discounted cumulative reward.

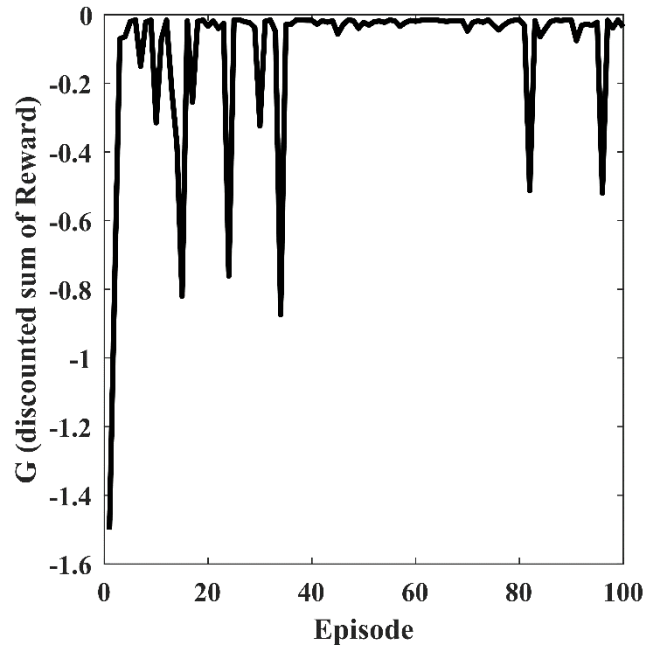


Figure 5. Cumulative Reward per Episode for one Learning Trial

In early episodes of learning, performance is poor both because exploration is intentionally high and because the agent has little knowledge of the system. After a relatively small number of episodes, however, the agent's knowledge of the system increases to a point where exploitation of that knowledge yields significantly higher rewards. During this period there is still significant exploration, though, and so some larger variances in reward can be observed. After around 50 episodes, as the rate of exploration continues to decrease, it becomes clear that the agent is consistently achieving high levels of reward in terms of NO_x concentration setpoint tracking in the presence of random disturbances.

From a control perspective, the performance of the agent at the first and last episodes, and with respect to the same disturbance signal, is shown in Figure 6, where k is the discrete timestep.

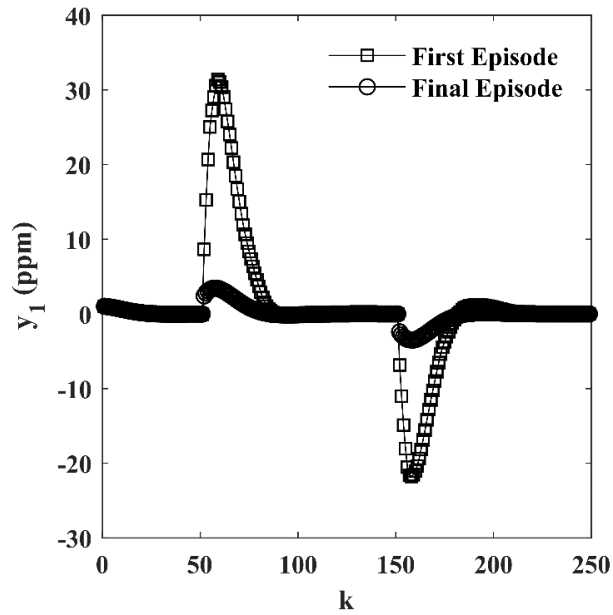


Figure 6. Output NOx Concentration Profiles for First and Last Episodes

Both cases exhibit zero offset corresponding to the structure of the underlying MPC. Performance in this case is clearly superior after episodic learning, with a ratio of ISE of 0.020 for the last episode to the first. Corresponding to this performance, Figure 7 shows the values of the parameters, namely the prediction and control horizons, selected by the RL agent during the final episode of operation.

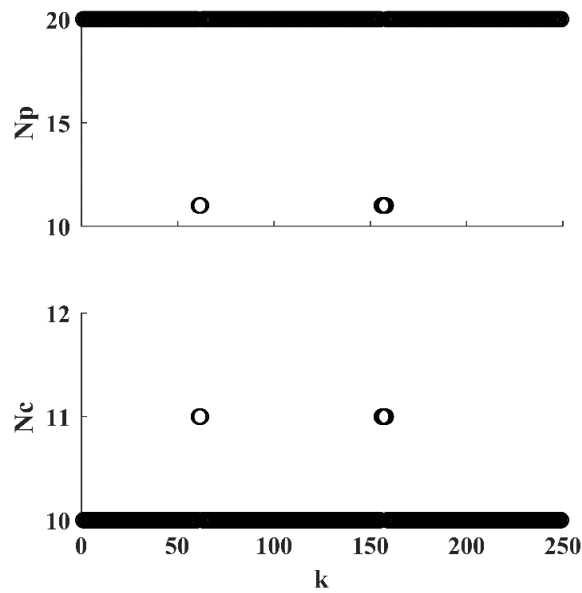


Figure 7. Parameters Selected During Last Episode

Where the output is close to zero, the same parameters are selected corresponding to proximity to similar basis function centers. However, at the points where the injected disturbance causes significant deviation, different tuning parameters are selected, leading to superior control.

Because the learning process requires random exploration of the state and action spaces, the same 100-episode trial conducted above was repeated 25 times to standardize results. Figure 8 shows the median cumulative reward for each episode over the 25 trials, along with the interquartile range (IQR).

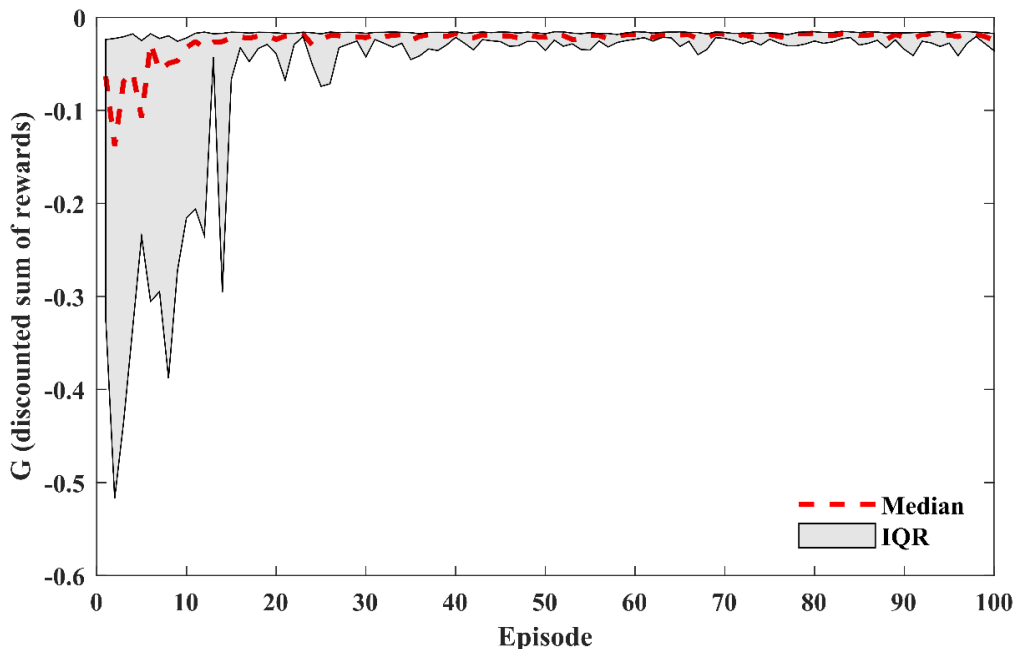


Figure 8. Cumulative Reward per Episode for 25 Trials

The key result from this figure is that there can be significant variation in the learning and control performance that is achieved across the initial episodes of operation. The median performance converges between the 20th and 30th episodes, and the lower quartile achieves a consistently high value around the 50th episode. Corresponding to this cumulative reward, the median ratio of ISE from the last episode to the first was 0.16 with lower and upper quartiles of 0.023 and 0.41, respectively. It should be noted that because Figure 5 shows the results from a specific trial that contained an outlier, a considerable decrease in G is seen even beyond 80 episodes while such decrease is not observed in IQR as presented in Figure 8.

Study on performance in response to unmodeled disturbance

In control of the SCR, disturbances on the flue gas temperature are common and can cause significant processes upsets. As such, a disturbance in the flue gas temperature was chosen to study the performance of the RL-MPC implementation in the presence of unmodeled disturbances. Figure 9(a, b) displays the

SCR output profiles for NO_x and ammonia concentrations for each of the controllers across the whole span of Study 1. Figure 9(c,d) similarly displays the input profile for ammonia injection flow and the disturbance signal for flue gas inlet temperature. Table 3 displays the ISE and IAE for each controller across the entire study along with its ratios to the ISE and IAE for the FBAFF controller representing the industrial comparison.

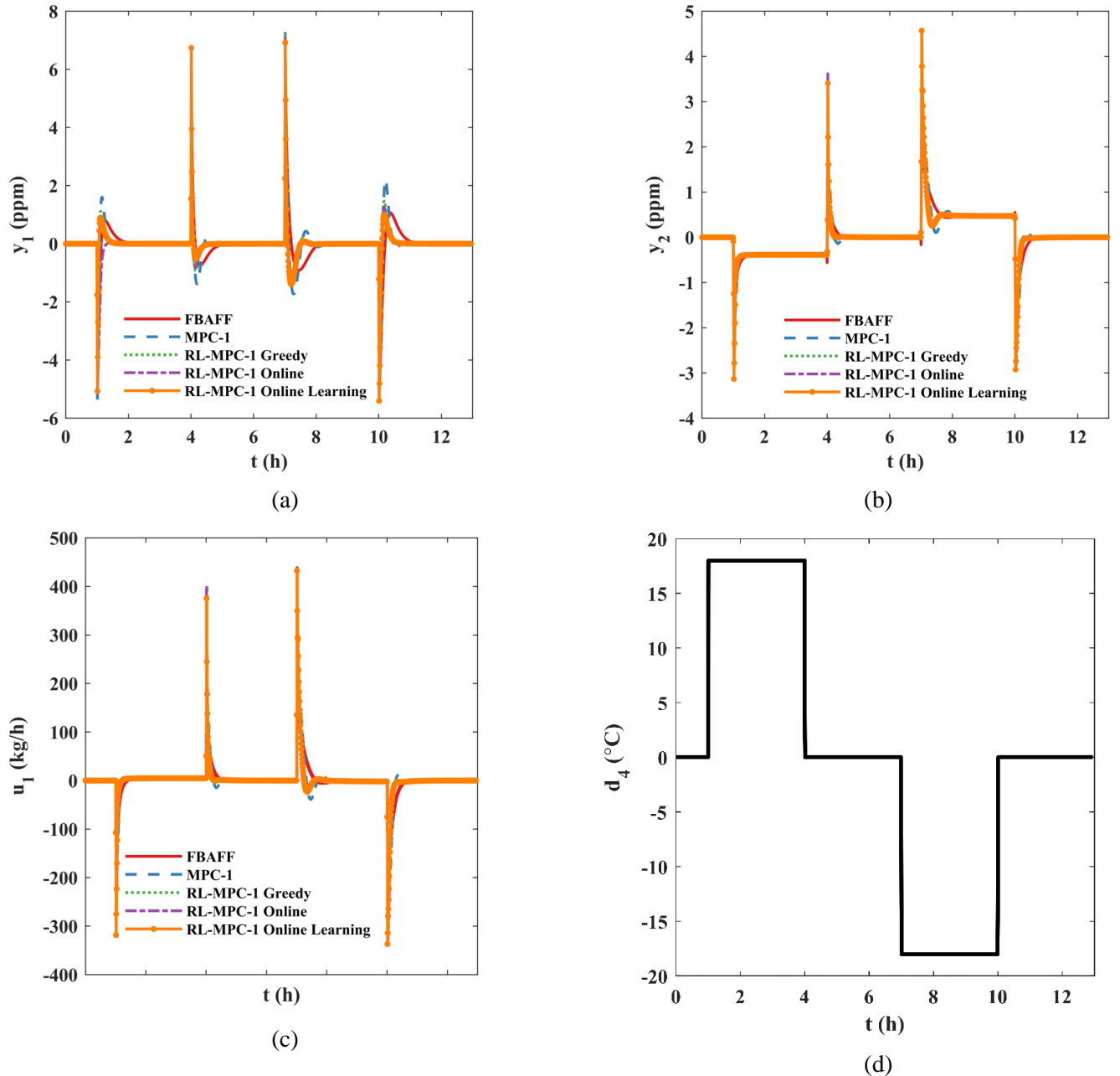
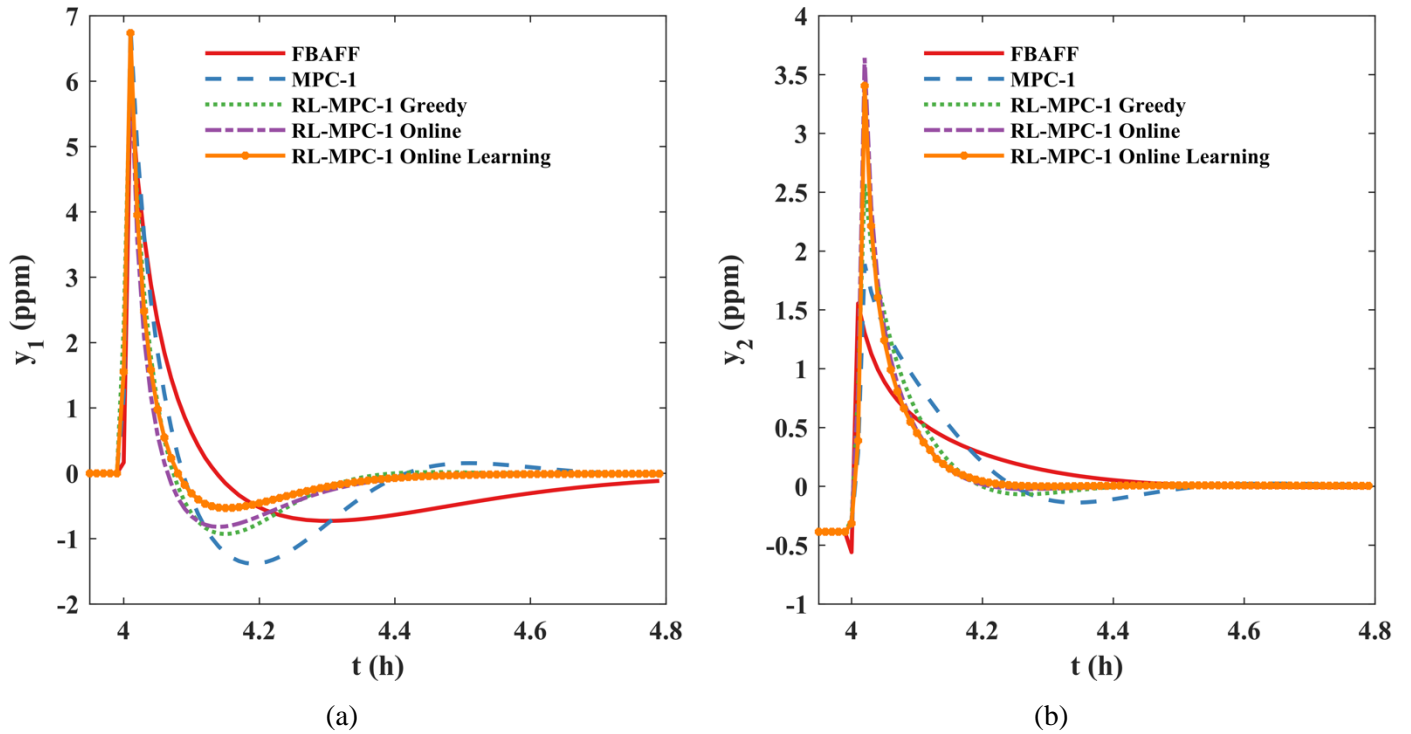


Figure 9. (a) Response for y_1 for disturbance rejection study, (b) Response for y_2 for disturbance rejection study, (c) Input profile for disturbance rejection study, (d) Disturbance profile for disturbance rejection study

Table 3. Summary of ISE and IAE for disturbance rejection study

Controller	ISE	Ratio to FBAFF	IAE	Ratio to FBAFF
FBAFF	622	--	300	--
MPC-1	685	1.10	276	0.92
RL-MPC-1 Greedy	462	0.74	255	0.85
RL-MPC-1 Online	453	0.73	250	0.83
RL-MPC-1 Online Learning	384	0.62	241	0.80

For ease of qualitative analysis, Figure 10 shows a zoomed in profile about the disturbance at hour 4 of the study.



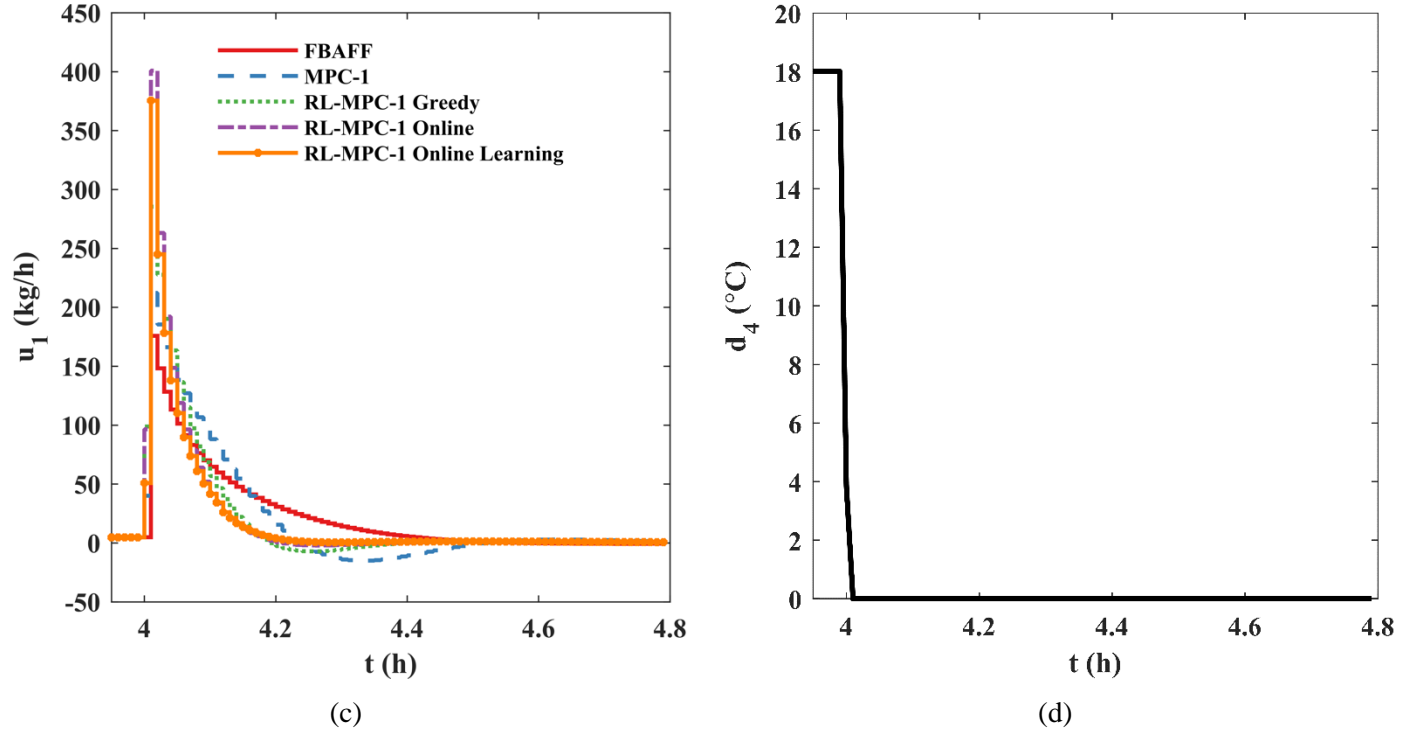


Figure 10. (a) Response for y_1 for disturbance rejection study at hour 4, (b) Response for y_2 for disturbance rejection study at hour 4, (c) Input profile for disturbance rejection study at hour 4, (d) Disturbance profile for disturbance rejection study at hour 4

Here it can be observed that each controller has a similar maximum deviation from the setpoint, but that the performance after this point varies significantly. The FBAFF does not perform well after the initial injection of the disturbance, settling more slowly to the setpoint than the other controllers. The MPC-1 without RL responds less aggressively than those with it, as reflected in the input profile. Standalone MPC-1 approaches the setpoint more quickly than FBAFF, but then overshoots and oscillates for a longer period, resulting in inferior performance with respect to ISE. Selecting actions greedily without learning online also results in inferior performance as compared to the best case, though RL-MPC-1 Greedy does perform better than the FBAFF controller and the static MPC-1. Finally, online learning (RL-MPC-1 Online) shows clearly improved performance, responding most aggressively at the onset of the disturbance and subsequently minimizing undershoot when returning to zero deviation. Performance of the controller (RL-MPC-1 Online Learning) can be seen to slightly improve after further learning is carried out online (as opposed to considering a fixed control law based on offline learning as in RL-MPC-1 Greedy), but this is not guaranteed because of the presence of some exploration while continuing to learn online.

Load-Following Study

To study the performance of the proposed RL-MPC-1 controllers under the effect of load following, a study was conducted by perturbing each of the five disturbance variables simultaneously along a profile mimicking load following for an SCR operating in a real coal-fired power plant. Here, again performance was evaluated following the same sequence of implementation and starting from the same offline weights as the flue gas inlet temperature disturbance rejection study above. Figure 11 (a,b) shows the output NO_x concentration (y_1) and ammonia concentration (y_2) profiles for each of the controllers, with the respective ISE and IAE measures reported in Table 4.

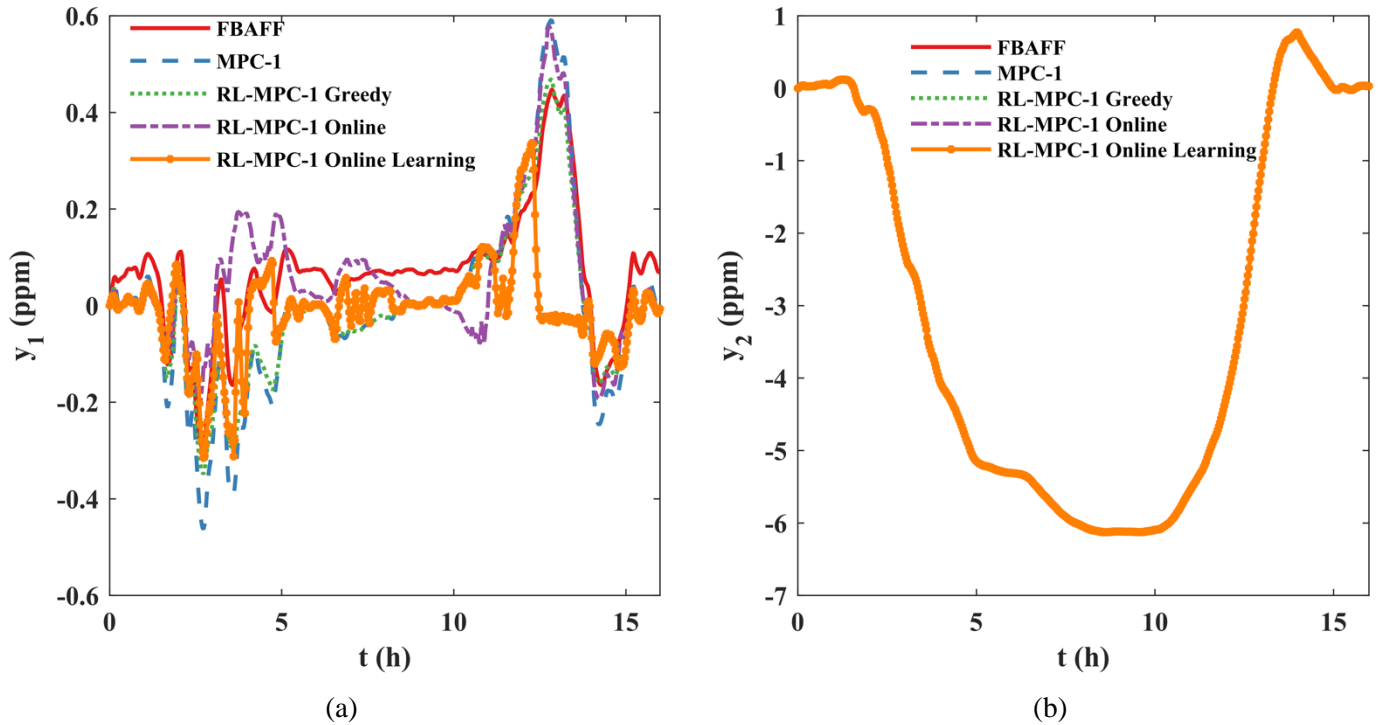


Figure 11. (a) Response for y_1 for load following study, (b) Response for y_2 for load following study

Table 4. Summary of ISE and IAE for load following study

Controller	ISE	Ratio to FBAFF	IAE	Ratio to FBAFF
FBAFF	33	--	175	--
MPC-1	59	1.8	203	1.16
RL-MPC-1 Greedy	38	1.2	164	0.94
RL-MPC-1 Online	40	1.1	162	0.93
RL-MPC-1 Online Learning	14	0.37	94	0.54

The first point to note here is that, contrary to the first study, the RL-MPC-1 controller does not immediately outperform other controllers. This is true because, while learning could have been conducted offline for the same sequence of load-following disturbances and the weights of the RL agent initialized with respect to only this operating scenario, the agent was instead initialized with random disturbances and initial states to span the entire expected operating range. Taken in conjunction with the small deviations in the output for this study because of the slow changes in the disturbance variables, the agent (RL-MPC-1 Online) learns for relatively little time in this operating region, yielding worse, though still acceptable, initial performance. As online operation progresses, however, the RL agent (RL-MPC-1 Online Learning) learns to control better in this area of the operating space and again outperforms the conventional controllers.

This effect can be better seen considering the ramp back up to the nominal condition between times 10 and 14 hours in Figure 12 for the outlet NO_x concentration.

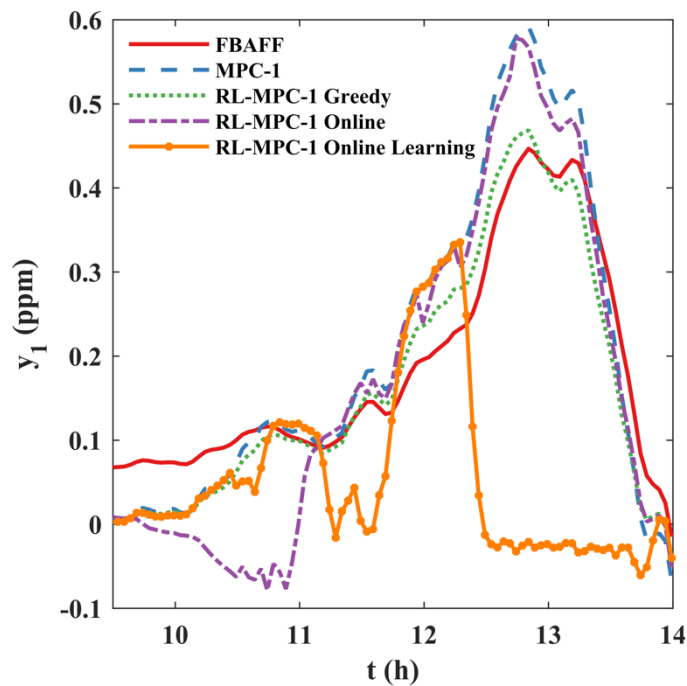


Figure 12. Response of y_1 during ramp up

Looking at the plot of the output, it is clear that the RL-MPC-1 after online operation learns a better set of tuning parameters to achieve better control. This is not necessarily clear through the plot of the input as the values are tightly bunched, but the input produced by the controller after some online learning produces slightly higher inputs faster than the other controllers, keeping the output closer to the setpoint throughout the ramp up to the nominal load. The effect of the time spent learning online can be clearly

seen when observing the tuning parameters, namely the control (N_C) and prediction (N_P) horizons, that are selected by the RL agent, as shown in Figure 13.

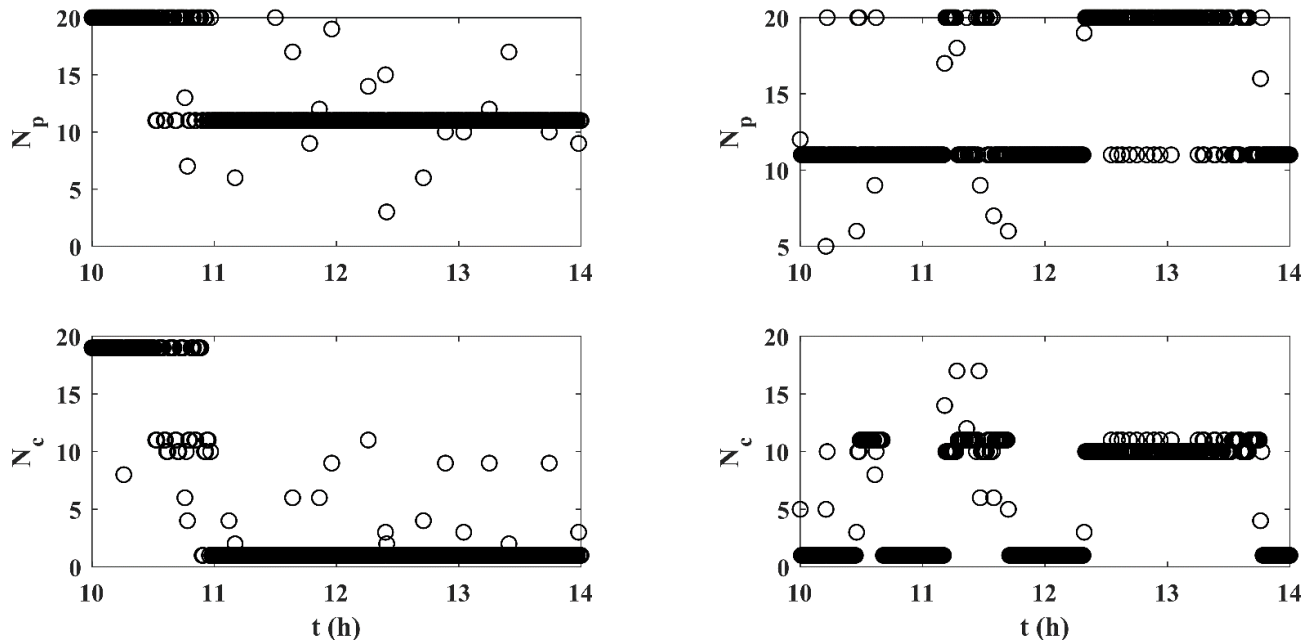


Figure 13. (a) Selected parameters during ramp up at first implementation, (b) Selected parameters during ramp up after online learning

The left-hand plot shows the parameters that are selected directly after the agent is implemented on the system, with the right showing the parameters selected after learning has continued for some time (it should be noted that some of the occasional deviations correspond to the 5% exploration rate for online learning, rather than direct differences in the RL agent). The parameters selected after initialization offline are close to constant, following with the idea that the agent has not had sufficient time in this operating space to learn. Clearly though, the RL agent adapts the tuning parameters to those needed to limit the output deviation in this region with relatively few samples needed following initialization offline. This is one of the key advantages of RL for process control - when encountering the need for further learning the agent can readily adapt online with no human intervention. The use of radial basis functions for function approximation is also important here, as the learning in this operating space only minimally affects the valid offline learning achieved in operating spaces further from zero deviation.

Study 2: MPC-2

To evaluate the performance of the second proposed MPC, a study was conducted to perturb the flue gas inlet temperature (d_4) to the SCR by +5% under three controllers. In addition to the baseline FBAFF controller, the other two are both implementations of MPC-2, one with a higher NO_x limit (MCP-2 High

Limit) and a zero-deviation NO_x limit (MPC-2 Low Limit). Figure 14 (a-d) shows the output and input profiles from this study.

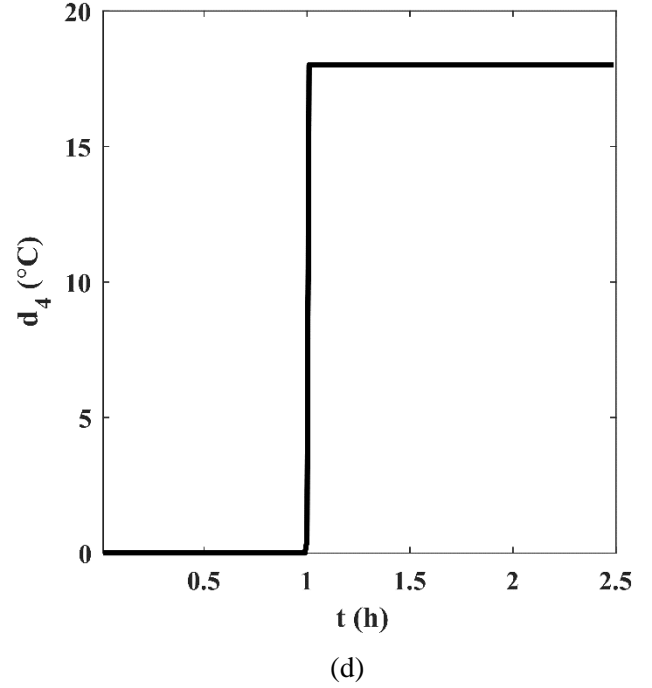
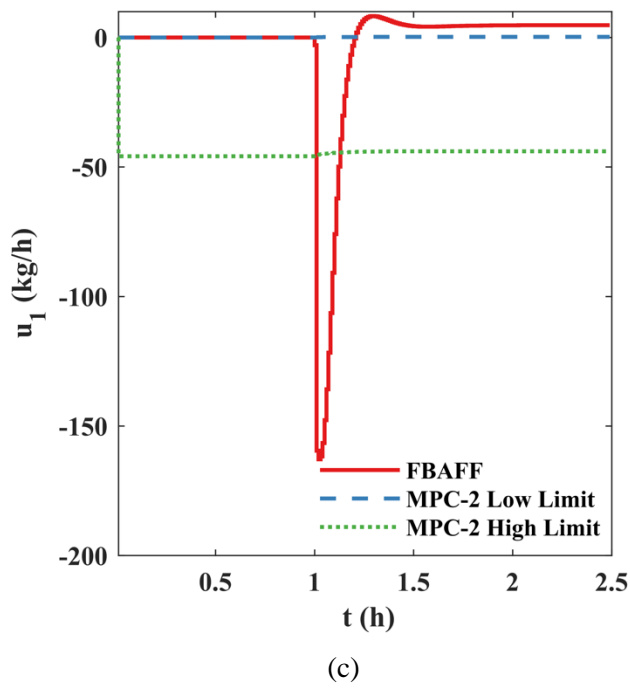
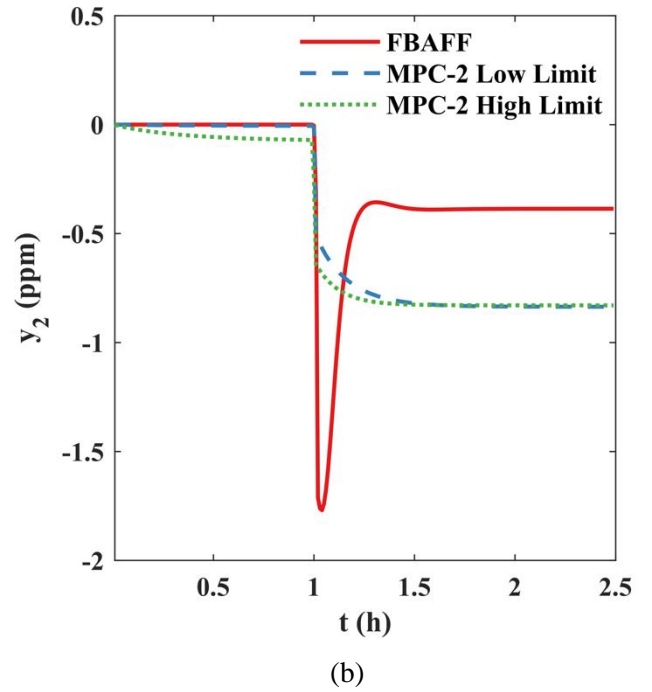
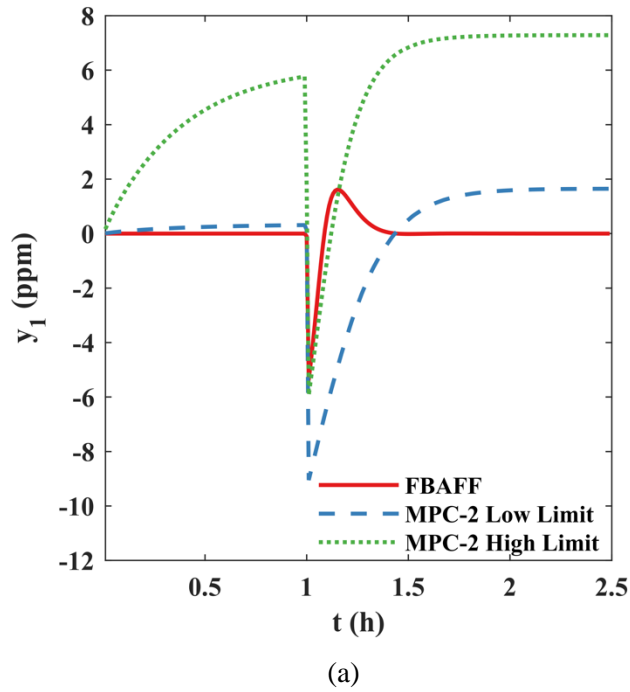


Figure 14. (a) Response for y_1 for disturbance rejection study using MPC-2, (b) Response for y_2 for disturbance rejection study using MPC-2, (c) Input profile for disturbance rejection study using MPC-2, (d) Disturbance profile for disturbance rejection study using MPC-2

Given the dual objective of the controller in this study, consideration is given to both the NO_x and NH_3 outlet concentrations in the flue gas. For the outlet NO_x concentration (y_1), it can be seen that, at the start of the study, the implementation of MPC-2 allowing for a higher outlet NO_x concentration (MPC-2 High Limit) does slightly reduce the amount of NH_3 being fed to the reactor (u_1) corresponding to a lower outlet NH_3 concentration (y_2). For this controller, an initial step change in inlet ammonia flow (u_1) is made to minimize the ammonia slip; this is reflected in the dynamic responses of both outputs before the onset of the disturbance. At the onset of the disturbance at $t=1h$, it can be seen that the FBAFF controller responds aggressively to limit the deviation of the NO_x concentration from its nominal value ($y_1=0$), while both instances of MPC-2 do not respond quickly, allowing the outlet NO_x to drift away from the nominal value in order to minimize the NH_3 at the outlet (y_2). This can be seen in the final values of both outputs, where the FBAFF controller returns to zero deviation, but both MPC-2 implementations yield some deviation from the nominal NO_x concentration to allow for minimization of the NH_3 slip in the new state. The implementation of a soft constraint of zero-deviation from the nominal load for MPC-2 (0 PPM) reduces the total NH_3 consumed by 0.94% when compared to the FBAFF controller, while allowing the NO_x to deviate slightly more provides a greater reduction of 5.4%.

5. Conclusions

A novel reinforcement learning algorithm for online tuning of model predictive controllers was developed and applied to an industrial control problem with challenging nonlinearities and time delay. The RL-augmented MPC algorithm is flexible and can be applied to any underlying MPC rather than requiring a specific formulation. The RL agent uses a two-stage approach whereby initialization is performed via offline learning on the controller model before deployment on the actual plant, with due consideration of controller objectives and constraints. The efficacy of the proposed algorithm was tested on an industrial SCR unit control application using a detailed one-dimensional SCR model developed in this study to account more completely for the complex dynamics. An industrial-standard SCR control implementation was compared to two RL-augmented MPCs – one for control of the outlet NO_x concentration and one used to minimize the ammonia slip at the outlet of the reactor while maintaining the NO_x concentration within some bounds. The first RL-MPC showed significant reduction in tracking error when compared to the standard control implementation. Results for offline learning were presented and discussed from the perspective of convergence of the cumulative reward of the RL agent across learning episodes. The results showed that continued learning online with a small percentage of exploration quickly yields better

performance than selecting actions based solely on offline learning. The second RL-augmented MPC showed good performance with respect to ammonia slip, reducing ammonia consumption in the SCR by about 1% when considering the nominal condition as the NO_x limit and around 5% when allowing a slight deviation from the nominal condition. The study demonstrates the promise of online application of RL-MPC not only for systems where optimal MPC tuning parameters change with time and/or servo control and/or disturbance rejection tasks, but also for systems where it is hard to obtain optimal tuning parameters in general without significant trial-and-error that may not be acceptable to plant personnel. While RL is used here to learn and adapt only the control horizons, RL can also be applied to adapt other MPC parameters such as weights and model parameters, but would require additional considerations.

Acknowledgements

This work was conducted as part of the Transformative Power Generation Program within the U.S. Department of Energy's Office of Fossil Energy and Carbon Management through the National Energy Technology Laboratory under the Mission Execution and Strategic Analysis contract (DE-FE0025912).

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. KeyLogic Systems, Inc.'s contributions to this work were funded by the National Energy Technology Laboratory under the Mission Execution and Strategic Analysis contract (DE-FE0025912) for support services. This material is based upon work supported by the U.S. Department of Energy, Office of Science, under Contract No. DEAC02-05CH11231.

Nomenclature

Symbol	Description	Units
J	Objective function	
A, B, C	Linear model parameters	
u	Input vector	
y	Output vector	
x	Control state vector	
d	Disturbance vector	
H, Q, D, M	Diagonal weighting matrices	
N_p	Prediction horizon	
N_c	Control horizon	
p	Output soft constraint	
ω	Minimum outlet NH ₃ concentration	
w	Weight vector	
\hat{q}	Quality function estimate	
R	Reward	
\bar{R}	Average reward	
N	Number	
G	Discounted cumulative reward	
I	Identity	
sp	Setpoint or setpoint trajectory	
t	Time	
P	Randomly drawn probability	
A	Set of all actions	
c	Basis function centroid	
q	Basis function	
n	Number of RL features (3)	
ISE	Integral squared error	
IAE	Integral absolute error	
a	Specific surface area of catalyst	m^2/m^3
C_i	Concentration of i	mol/m^3
C_{is}	Surface concentration of i	mol/m^3
d	Catalyst hydraulic diameter	m

d_{cat}	Catalyst pitch	m
D_H	Hydraulic diameter	m
D_{ij}	Binary diffusivity of component i in j	m^2/h
$D_{eff,ij}$	Effective diffusivity of component i in j	m^2/h
$D_{mix,i}$	Mixture diffusivity of component i in mixture	m^2/h
E_{des}^0	Desorption activation energy at 280°C	$kcal/mol$
E_{des}	Desorption activation energy	$kcal/mol$
E_{NO}	Reduction activation energy	$kcal/mol$
E_{ox}	Oxidation activation energy	$kcal/mol$
k_{ads}	Adsorption pre-exponential factor	$m^3/mol \cdot h$
k_{des}	Desorption pre-exponential factor	h^{-1}
k_{NO}	Reduction pre-exponential factor	h^{-1}
k_{ox}	Oxidation pre-exponential factor	h^{-1}
k_g	External mass transfer coefficient	m/h
L_{cat}	Length of catalyst	m
n_i	Molar flowrate of species i	mol/h
n_{tot}	Total molar flowrate	mol/h
r_{ads}	Adsorption reaction rate	h^{-1}
r_{des}	Desorption reaction rate	h^{-1}
r_{NO}	Reduction reaction rate (the main or desired reaction)	h^{-1}
r_{ox}	Oxidation reaction rate (the side or undesired reaction)	h^{-1}
Re	Reynolds number	
Sc	Schmidt number	
Sh	Sherwood number	
z_i	Mole fraction of i	

Greek

α	RL Step Size	
β	Average Reward Step Size	
γ	Discount Factor	
Ψ	Regression Vector	
Φ	Parameter Vector	
ε	Exploration Rate	
δ	Temporal Difference	
σ	Gaussian shape parameter	
α	Surface coverage dependence parameter	---
ε_b	Bed porosity	---
θ_B	Surface coverage fraction of B	---
μ	Viscosity	$kg/m \cdot h$
ν	Kinematic viscosity	m^2/h
ρ	Density	kg/m^3
ρ_c	Catalyst density	kg/m^3
ϕ_1	Thiele modulus	---
Ω_B	Catalyst adsorption capacity	$mol NH_3/m^3$
τ	Tortuosity	
η_{int}	Internal effectiveness factor	

Subscripts/Superscripts

Description

k	Discrete time instant
i	Controller variable index
lb	Lower bound
ub	Upper bound
u	Unmeasured
m	Measured
*	Optimal
ep	Episode

Bibliography

- Beeckman, J.W., Hegedus, L.L., 1991. Design of Monolith Catalysts for Power Plant NO_x Emission Control. *Industrial and Engineering Chemistry Research* 30, 969–978. <https://doi.org/10.1021/ie00053a020>
- Beretta, A., Orsenigo, C., Ferlazzo, N., Tronconi, E., Forzatti, P., Berti, F., 1998. Analysis of the performance of plate-type monolithic catalysts for selective catalytic reduction DeNO_x applications. *Industrial and Engineering Chemistry Research* 37, 2623–2633. <https://doi.org/10.1021/ie970791m>
- Bhattacharyya, D., Rengaswamy, R., 2010. System Identification and Nonlinear Model Predictive Control of a Solid Oxide Fuel Cell. *Industrial & Engineering Chemistry Research* 49, 4800–4808. <https://doi.org/https://doi.org/10.1021/ie9020254>
- Bruneji, L.A., Lee, J.M., Shah, S.L., 2010. Dynamic tuning of PI-controllers based on model-free Reinforcement Learning methods, in: *Proceedings - International Conference on Control, Automation and Systems*. pp. 453–458. <https://doi.org/10.1109/ICCAS.2010.5669655>
- Carlucho, I., De Paula, M., Villar, S., Acosta, G.G., 2017. Incremental Q-learning strategy for adaptive PID control of mobile robots. *Expert Systems with Applications* 80, 183–199. <https://doi.org/10.1016/j.eswa.2017.03.002>
- Coal - DNX HD - SCR DeNO_x catalyst DNX®-series, 2020.
- Estimating Ammonia Emissions from Stationary Power Plants, 2009. . Palo Alto, CA.
- Fogler, H.S., 2006. *Elements of Chemical Reaction Engineering*. Prentice Hall, Upper Saddle River, NJ.
- Görges, D., 2017. Relations between Model Predictive Control and Reinforcement Learning. *IFAC-PapersOnLine* 50, 4920–4928. <https://doi.org/10.1016/j.ifacol.2017.08.747>
- Gros, S., Zanon, M., 2020. Data-driven economic NMPC using reinforcement learning. *IEEE Transactions on Automatic Control* 65, 636–648. <https://doi.org/10.1109/TAC.2019.2913768>
- Kamthe, S., Deisenroth, M.P., 2018. Data-efficient reinforcement learning with probabilistic model predictive control, in: *International Conference on Artificial Intelligence and Statistics, AISTATS 2018*. PMLR, pp. 1701–1710.
- Kanniche, M., Mourigal, M., Gros, S., 2011. Modeling of NO_x Selective Catalytic Reduction in pulverised coal power plants, in: *Chemical Engineering Transactions*. Italian Association of Chemical Engineering - AIDIC, pp. 671–676. <https://doi.org/10.3303/CET1125112>
- Kim, Y., Lee, J.M., 2020. Model-based reinforcement learning for nonlinear optimal control with practical asymptotic stability guarantees. *AIChE Journal* 66. <https://doi.org/10.1002/aic.16544>
- Konidaris, G., Csail, M., Osentoski, S., Thomas, P., 2011. Value Function Approximation in Reinforcement Learning using the Fourier Basis. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* 380–385.
- Leopold, T., 2010. Tuning Ammonia Flow to Optimize SCR Performance. *Power*.

- Lietti, L., Nova, I., Camurri, S., Tronconi, E., Fonatti, P., 1997. Dynamics of the SCR-DeNO_x Reaction by the Transient-Response Method. *AIChE Journal* 43, 2559–2570.
- Lv, Y., Liu, J., Yang, T., Zeng, D., 2013. A novel least squares support vector machine ensemble model for NO_x emission prediction of a coal-fired boiler. *Energy* 55, 319–329. <https://doi.org/10.1016/j.energy.2013.02.062>
- Lv, Y., Lv, X., Fang, F., Yang, T., Romero, C.E., 2020. Adaptive selective catalytic reduction model development using typical operating data in coal-fired power plants. *Energy* 192. <https://doi.org/10.1016/j.energy.2019.116589>
- Mayne, D.Q., 2014. Model predictive control: Recent developments and future promise. *Automatica* 50, 2967–2986. <https://doi.org/10.1016/j.automatica.2014.10.128>
- Miccio, M., Cosenza, B., 2014. Control of a distillation column by type-2 and type-1 fuzzy logic PID controllers. *Journal of Process Control* 24, 475–484. <https://doi.org/10.1016/j.jprocont.2013.12.007>
- Mitchell, S.C., 1998. NO_x in pulverised coal combustion, CCC/05.
- Mobed, P., Maddala, J., Rengaswamy, R., Bhattacharyya, D., Turton, R., 2014. Data reconciliation and dynamic modeling of a sour water gas shift reactor, in: *Industrial and Engineering Chemistry Research*. American Chemical Society, pp. 19855–19869. <https://doi.org/10.1021/ie500739h>
- Morinelly, J.E., Ydstie, B.E., 2016. Dual MPC with Reinforcement Learning. 11th IFAC Symposium on Dynamics and Control of Process Systems 266–271. <https://doi.org/https://doi.org/10.1016/j.ifacol.2016.07.276>
- Mowbray, M., Smith, R., Del Rio- Chanona, E.A., Zhang, D., 2021. Using process data to generate an optimal control policy via apprenticeship and reinforcement learning. *AIChE Journal* 1–15. <https://doi.org/10.1002/aic.17306>
- Muñoz, E., Marín, P., Ordóñez, S., Díez, F. v., 2015. The role of reaction kinetics and mass transfer in the selective catalytic reduction of NO with NH₃ in monolithic reactors. *Journal of Chemical Technology and Biotechnology* 90, 1299–1307. <https://doi.org/10.1002/jctb.4437>
- Nian, R., Liu, J., Huang, B., 2020. A review On reinforcement learning: Introduction and applications in industrial process control. *Computers and Chemical Engineering* 139. <https://doi.org/10.1016/j.compchemeng.2020.106886>
- Nova, I., Beretta, A., Groppi, G., Lietti, L., Tronconi, E., Forzatti, P., 2005. Monolithic Catalysts for NO_x Removal from Stationary Sources, in: Cybulski, A., Moulijn, J.A. (Eds.), *Structured Catalysts and Reactors*. CRC Press, pp. 171–214. <https://doi.org/10.1201/9781420028003.ch6>
- Nova, I., Lietti, L., Tronconi, E., Forzatti, P., 2000. Dynamics of SCR reaction over a TiO₂-supported vanadia-tungsta commercial catalyst. *Catalysis Today* 60, 73–82. [https://doi.org/https://doi.org/10.1016/S0920-5861\(00\)00319-9](https://doi.org/https://doi.org/10.1016/S0920-5861(00)00319-9)
- Ogunnaike, B.A., Ray, W.H., 1994. *Process Dynamics, Modeling, and Control*. Oxford University Press, New York.
- Pan, T., Li, S., Cai, W.J., 2007. Lazy learning-based online identification and adaptive PID control: A case study for CSTR process. *Industrial and Engineering Chemistry Research* 46, 472–480. <https://doi.org/10.1021/ie0608713>

- Peng, H., Shioya, H., Zou, R., 2006. A predictive control strategy for nonlinear NO_x decomposition process in thermal power plants. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans* 36, 904–921. <https://doi.org/10.1109/TSMCA.2005.855920>
- Petsagkourakis, P., Sandoval, I.O., Bradford, E., Zhang, D., del Rio-Chanona, E.A., 2020. Reinforcement learning for batch bioprocess optimization. *Computers and Chemical Engineering* 133. <https://doi.org/10.1016/j.compchemeng.2019.106649>
- Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. *Control Engineering Practice* 11, 733–764. [https://doi.org/https://doi.org/10.1016/S0967-0661\(02\)00186-7](https://doi.org/https://doi.org/10.1016/S0967-0661(02)00186-7)
- Qin, T., Yang, T., Lv, Y., Hong, F., Yao, Q., 2016. Dynamic modeling for SCR system of coal fired power plant, in: *Chinese Control Conference, CCC*. IEEE Computer Society, Chengdu, China, pp. 2006–2010. <https://doi.org/10.1109/ChiCC.2016.7553660>
- Reid, R.C., Prausnitz, J.M., Poling, B.E., 1987. *The properties of gases and liquids*, Fourth. ed. McGraw-Hill, New York, New York, USA.
- Safdarnejad, S.M., Tuttle, J.F., Powell, K.M., 2019. Dynamic modeling and optimization of a coal-fired utility boiler to forecast and minimize NO_x and CO emissions simultaneously. *Computers and Chemical Engineering* 124, 62–79. <https://doi.org/10.1016/j.compchemeng.2019.02.001>
- Shah, H., Gopal, M., 2016. Model-free predictive control of nonlinear processes based on reinforcement learning. *IFAC-PapersOnLine* 49, 89–94. <https://doi.org/10.1016/j.ifacol.2016.03.034>
- Shah, S., Abrol, S., Balram, S., Barve, J., 2015. Optimal Ammonia Injection for Emissions Control in Power Plants. *IFAC-PapersOnLine* 48, 379–384. <https://doi.org/10.1016/j.ifacol.2015.12.408>
- Shen, B., Zhao, N., Liu, T., Wu, F., Zuo, C., 2012. Modeling and simulation of selective catalytic reduction for flue gas denitration in power plants, in: *Advanced Materials Research*. pp. 6580–6586. <https://doi.org/10.4028/www.scientific.net/AMR.383-390.6580>
- Shin, J., Badgwell, T.A., Liu, K.H., Lee, J.H., 2019. Reinforcement Learning – Overview of recent progress and implications for process control. *Computers and Chemical Engineering* 127, 282–294. <https://doi.org/10.1016/j.compchemeng.2019.05.029>
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driessche, G., Graepel, T., Hassabis, D., 2017. Mastering the game of Go without human knowledge. *Nature* 550, 354–359. <https://doi.org/10.1038/nature24270>
- Spielberg, S., Tulsyan, A., Lawrence, N.P., Loewen, P.D., Gopaluni, R.B., 2019. Towards Self-Driving Processes: A Deep Reinforcement Learning Approach to Control. *AICHE Journal*. <https://doi.org/10.1002/aic.16689>
- Sutton, R.S., Barto, A.G., 2018. *Reinforcement Learning: An Introduction*, 2nd ed. Bradford, Cambridge, MA, USA.
- Uberoi, M., Pereira, C.J., 1996. External Mass Transfer Coefficients for Monolith Catalysts. *Industrial and Engineering Chemistry Research* 35, 113–116. <https://doi.org/10.1021/ie9501790>

- Yoo, H., Kim, B., Kim, J.W., Lee, J.H., 2021. Reinforcement learning based optimal control of batch processes using Monte-Carlo deep deterministic policy gradient with phase segmentation. *Computers and Chemical Engineering* 144. <https://doi.org/10.1016/j.compchemeng.2020.107133>
- Zanon, M., Gros, S., Bemporad, A., 2019. Practical reinforcement learning of stabilizing economic MPC, in: 2019 18th European Control Conference, ECC 2019. Institute of Electrical and Electronics Engineers Inc., pp. 2258–2263. <https://doi.org/10.23919/ECC.2019.8795816>
- Zhang, K., Zhao, J., Zhu, Y., 2018. MPC case study on a selective catalytic reduction in a power plant. *Journal of Process Control* 62, 1–10. <https://doi.org/10.1016/j.jprocont.2017.11.010>
- Zhang, K., Zhao, J., Zhu, Y., Xu, Z., 2016. Model predictive control case study: Selective catalytic reduction (SCR) system in coal-fired power plant, in: Chinese Control Conference, CCC. IEEE Computer Society, pp. 4300–4305. <https://doi.org/10.1109/ChiCC.2016.7554021>

Appendix A – Details on SCR Mass Transfer Modeling

The external mass transfer is calculated from the following relationships, derived from a combination of expressions from Munoz et al. (Muñoz et al., 2015) with the kinetic expressions previously described (Lietti et al., 1997; Muñoz et al., 2015; Nova et al., 2000). The interphase equations governing the relationship between the surface concentration and the bulk concentrations of NO and NH₃ relate these concentrations to the bed porosity (ε_b), adsorption capacity of the catalyst (Ω_{NH_3}), and the internal effectiveness of the catalyst (η_{int}) as well as the rates of the reactions taking place on the catalyst surface.

$$ak_{g,NO}(C_{NO,s} - C_{NO}) = -\Omega_{NO}\eta_{int}r_{red}(1 - \varepsilon_b) \quad (30)$$

$$ak_{g,NH_3}(C_{NH_3,s} - C_{NH_3}) = -\Omega_{NH_3}\eta_{int}(r_{ads} - r_{des})(1 - \varepsilon_b) \quad (31)$$

The internal effectiveness factor is calculated using Equation (30). (Fogler, 2006) The binary diffusion coefficients (D_{ij}) are called in ACM using temperature, pressure, and mole fraction as parameters and implementing the Chapman-Enskog-Wilke-Lee model to calculate their values. (Reid et al., 1987)

$$\eta_{int} = \frac{3}{\phi_1^2} (\phi_1 \coth \phi_1 - 1) \quad (32)$$

$$\phi_1 = R \sqrt{\frac{r_{red}}{D_{eff,NO-NH_3}}} \quad (33)$$

$$D_{eff,ij} = D_{ij}\varepsilon_b/\tau \quad (34)$$

Equations for the mixture diffusivity were taken from Mobed et al. (Mobed et al., 2014)

$$D_{mix,i} = \frac{(1 - z_i)}{\sum_{j \neq i} \left(\frac{z_j}{D_{ij}} \right)} \quad (35)$$

The external mass transfer coefficient is calculated using Equation 34 based on the work of Uberoi and Pereira. (Uberoi and Pereira, 1996)

$$Sh = 2.696 * \left(1 + 0.139 * Sc * Re * \left(\frac{D_{H,cat}}{L_{cat}} \right) \right)^{0.81} \quad (36)$$

$$a = \frac{4\varepsilon_b}{s_{cat}} \quad (37)$$

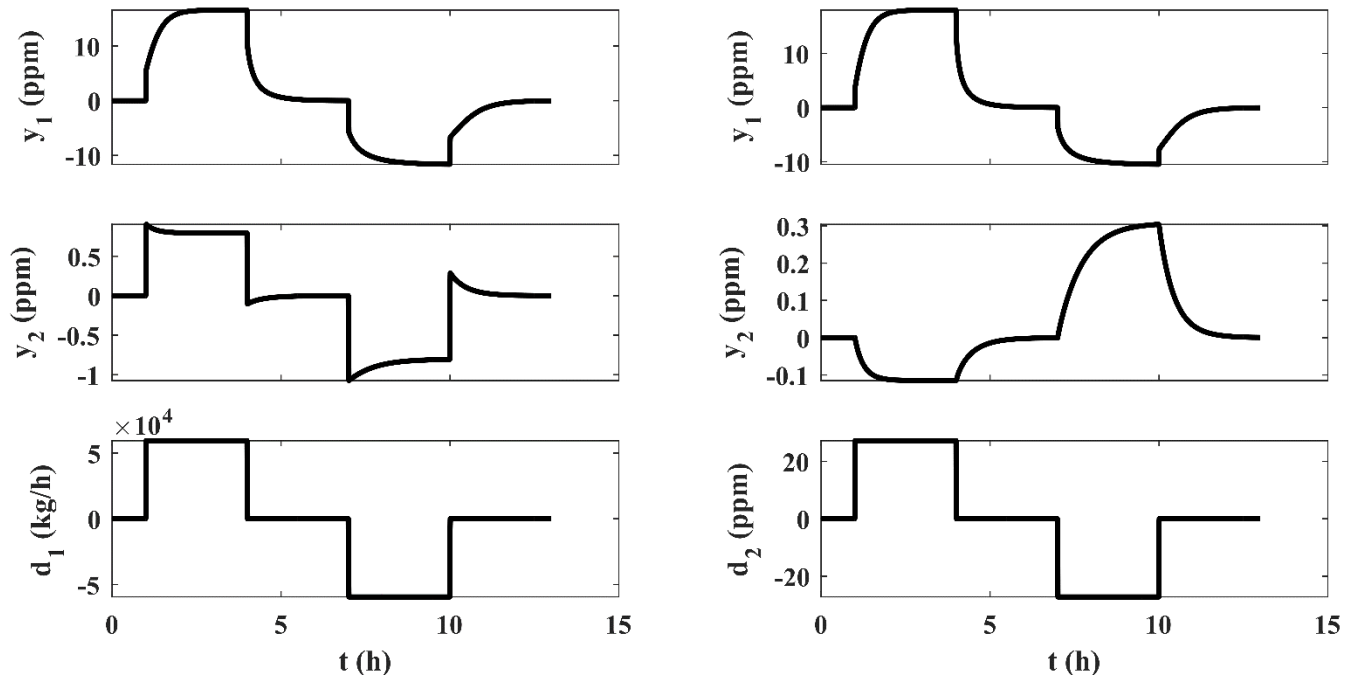
$$Sh = \frac{k_g D_{H,cat}}{D_{mix}} \quad (38)$$

$$Sc = \frac{\mu}{\rho D_{mix}} \quad (39)$$

$$Re = \frac{\rho D_{H,cat} v}{\mu} \quad (40)$$

Appendix B – Open Loop Control Responses

Figure 15 displays the responses to the other disturbances that were studied for this system.



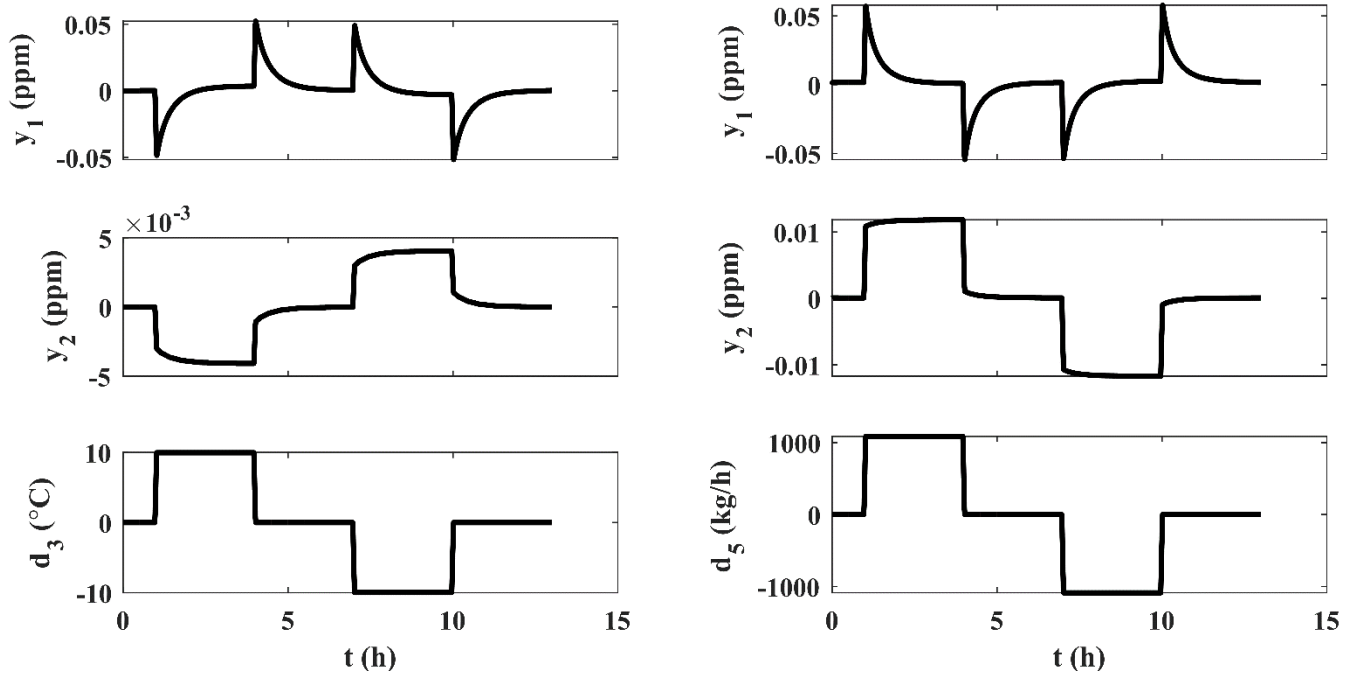


Figure 15. Open Loop Responses for Multiple Disturbances

Focusing on the outlet NO_x concentration (y_1), the responses to step changes in d_1 and d_2 are relatively large in magnitude (> 10 ppm) while the responses with respect to d_3 and d_5 are small (< 0.1 ppm). This trend holds for the outlet ammonia concentration (y_2).

Table 5 displays the open loop characteristics for each input or disturbance with respect to each output. For each variable, the input or disturbance has been perturbed by +10%.

Table 5. Open Loop Characterization of SCR Input and Disturbances

Variable	Gain (ppm)		Time Constant (h)	
	$K_1 (y_1)$	$K_2 (y_2)$	$\tau_1 (y_1)$	$\tau_2 (y_2)$
u_1	-7.683	1.217	0.335	0.05
d_1	16.57	0.794	0.240	0.05
d_2	18.01	-0.115	0.27	0.25
d_3	0.004	-0.004	0.28	0.01
d_4	1.322	-0.835	0.65	0.03
d_5	0.001	0.0118	0.9799	0.01

Appendix C – Summary of Reinforcement Learning Setup and Parameters

This appendix presents the setup of the RL agent while learning in the episodic learning mode on the reduced order model. Over the course of the learning studies the RL agent hyperparameters remained unchanged. Table 6 summarizes the setup of the RL agent. These values were obtained from a combination of standard literature values, manual tuning, and the dimensionality of the system. The initial step size parameter, α_0 , was taken to be as large as possible without causing convergence problems. Values of the discount parameter, γ , were commonly observed to be on $[0.9, 0.99]$ in the literature (Sutton and Barto, 2018). Minimum values of N_p and N_c are mathematically constrained, and the maximum values were determined by manual tuning. The scaling factors on the output indicate the expected range of deviation under normal operation, though deviation outside of this range is still allowable. The number of discretizations was determined empirically to balance the fidelity of the learning with computational performance. The number of timesteps in each episode was taken to be sufficiently long to capture the designed experiments for learning.

Table 6. Summary of RL Parameters

α_0	0.01
γ	0.95
ε_0	0.05
ε_1	0.95
$N_{p, \min}$	1
$N_{p, \max}$	20
$N_{c, \min}$	1
$N_{c, \max}$	20
y_{\min}	-10
y_{\max}	10
n	$\{y_1, N_p, N_c\} = 3$
$N_{\text{discretizations}}$	10
$N_{\text{Basis Functions}}$	$(N_{\text{discretizations}})^n = 1000$
c_i	Evenly distributed over scaled space
$N_{\text{timesteps, max}}$	250