Adaptive Model Predictive Batch Process Monitoring and Control

Masoud Kheradmandi and Prashant Mhaskar*

Department of Chemical Engineering, McMaster University, Hamilton, Ontario, Canada

E-mail: mhaskar@mcmaster.ca

Abstract

The present work addresses the problem of loss of model validity in batch process control via online monitoring and adaptation based model predictive control. To this end, a state space subspace-based model identification method suitable for batch processes is utilized and then a model predictive controller is designed. To monitor model performance, a model validity index is developed for batch processes. In the event of poor prediction (observed via breaching of a threshold by the model validity index), re-identification is triggered to identify a new model, and thus adapt the controller. In order to capture the most recent process dynamics, the identification is appropriately designed to emphasize more the recent process data. The efficacy of the proposed method is demonstrated using an electric arc furnace as a simulation test bed.

1 Introduction

Optimal operation of batch processes is essential to achieve consistent and high-quality products, and to avoid wasted batches. The operation and control of batch processes, however,

has to deal with involves numerous challenges such as nonlinearity, complex variable interactions and constraints.[1–5] One control method well suited to handling these challenges is model predictive control (MPC). In MPC, an optimization problem is solved at each sampling instance over a finite time horizon, subject to the dynamic model of the plant and process constraints, to compute the control action. Regardless of the nature of the model used (mechanistic or data driven), the resultant plant-model mismatch remains unavoidable.

One approach to address the plant model mismatch is through robust/offset free MPC design. In robust MPC approaches, the manipulated input action is calculated to handle the worst case scenario of the uncertainty.[6,7] In one formulation, Lyapunov-based stability constraint is utilized that enables explicit characterization of the robust stability region (region from where stability of the closed-loop system is guaranteed in the presence of constraints and uncertainty).[8] In another approach, plant model mismatch is handled by integrating disturbances by so-called offset-free MPC, to overcome offset in set-point tracking.[9,10] In one direction, Dynamic Mode Decomposition with control (DMDc) has been proposed to extract low-order models from high-dimensional, complex systems.[11,12]

While these control algorithms are designed to reject disturbances at steady state, the possibility of improved dynamic closed-loop performance motivates online model monitoring (and correction). In this direction, the main focus of most of the existing contributions was on

controller performance monitoring. Many controller performance assessment approaches are based on comparing the controller performance with an ideal benchmark.[13–15] The prescribed remedy for poor performance is controller parameter retuning. In another direction[16] model prediction performance is directly monitored for continuous processes. In the case of poor model performance re-identification is triggered to achieve model improvement.

There also exist other MPC approaches with re-identification (IMPC) built-in. In most of these results, excitation constraints are included to ensure that the recent data possesses sufficient richness for determining the model parameters. Early results using this approach[17] require identification at every time step. In later contributions,[18,19] input excitation constraint is implemented based on a trigger, activated on poor model prediction. In a recent contribution, in order to avoid additional constraints in MPC, previous data was also used in re-identification step with recent data of the plant. In this method, the controller looks back in time to evaluate model prediction performance. Then, in the case of poor prediction, the re-identification is triggered and earlier data and recent plant data are augmented for the purpose of model identification in continuous operation. In the context of batch process operation, the problem of model validity monitoring and re-identification, however, remains addressed. Note that every new batch goes through a learning phase, where a direct application of the ideas from[16] would result in unnecessary re-identification. Furthermore, the

results in[16] do not account for the need to emphasize current batch data more than previous data, and need to be appropriately adapted for batch processes.

Motivated by the above considerations, in this work we address the problem of model prediction performance monitoring and adaptive model for MPC design of batch processes. The rest of the manuscript is organized as follows: First, the general description for the batch systems considered in this work, a subspace identification approach and a representative formulation for linear MPC are reviewed. Then the proposed model monitoring, triggering and re-identification approach are presented. The efficiency of the proposed MPC with re-identification for batch processes is illustrated by implementation on a nonlinear electric arc furnace (EAF) simulation example. Finally, concluding remarks are presented.

2 Preliminaries

This section presents a brief description of the general class of processes that are considered in this manuscript, followed by a review of subspace identification and a conventional MPC formulation.

2.1 System Description

Consider a general multi-input multi-output (MIMO) batch process, with measured outputs denoted by $y \in \mathbb{R}^{n_y}$, and manipulated input variables denoted by $u \in \mathbb{R}^{n_u}$, taking values in a nonempty convex subset $\mathcal{U} \subset \mathbb{R}^{n_u}$, where $\mathcal{U} = \left\{u \in \mathbb{R}^{n_u} \mid u_{\min} \leq u \leq u_{\max}\right\}$, $u_{\min} \in \mathbb{R}^{n_u}$ and $u_{\max} \in \mathbb{R}^{n_u}$ denote the lower and upper bounds of the input variables. A discrete implementation of MPC is utilized, thus u is piecewise constant and defined over an arbitrary sampling instance k as:

$$u(t) = u(k), \;\; k\Delta t \leq t < (k+1)\Delta t$$

where Δt is the sampling time and x_k and y_k denote state and output at the kth sample time. We consider the case where the MPC is implemented based on a linear (identified) model, identified using subspace identification techniques, and address the problem of monitoring model quality online, and triggering re-identification as appropriate, to maintain model validity and closed-loop performance.

2.2 Subspace-Based Identification

In this section, a representative subspace based identification method is presented and reviewed.[20–22] The goal of the state space subspace identification methods is to determine the

linear time invariant model matrices for a discrete LTI model of the following form:

$$x_{k+1} = Ax_k + Bu_k \tag{1}$$

$$y_k = Cx_k + Du_k \tag{2}$$

where $x \in \mathbb{R}^{n_x}$ and $y \in \mathbb{R}^{n_y}$ denote the vector of state variables and measured outputs. In order to calculate A, B, C and D matrices, data-Hankel matrices are first constructed using the input and output variables as follows:

$$U_p = U_{1|i_H} = \begin{bmatrix} u_1 & u_2 & \dots & u_j \\ u_2 & u_3 & \dots & u_{j+1} \\ \dots & \dots & \dots & \dots \\ u_{i_H} & u_{i_H+1} & \dots & u_{i_H+j-1} \end{bmatrix} \tag{3}$$

$$U_f = U_{i_H+1|2i_H} = \begin{bmatrix} u_{i_H+1} & u_{i_H+2} & \dots & u_{i_H+j} \\ u_{i_H+2} & u_{i_H+3} & \dots & u_{i_H+j+1} \\ \dots & \dots & \dots & \dots \\ u_{2i_H} & u_{2i_H+1} & \dots & u_{2i_H+j-1} \end{bmatrix} \tag{4}$$

where U_p and U_f denote the past and future input Hankel matrices. i_H is a user-specified parameter that limits the order of the system (n) (which in itself is a user-specified parameter). Similarly block-Hankel matrices are defined for outputs where $Y_p, Y_f \in \mathbb{R}^{i_H n_y \times j}$ are defined similar to the input Hankel matrices. The state sequences matrices are defined as

follows:

$$X_p = \begin{bmatrix} x_1 & x_2 & \dots & x_j \end{bmatrix} \tag{5}$$

$$X_f = \begin{bmatrix} x_{i_H+1} & x_{i_H+2} & \dots & x_{i_H+j} \end{bmatrix} \tag{6}$$

furthermore, the following variables are used in the approach:

$$\Psi_p = \begin{bmatrix} Y_p \\ U_p \end{bmatrix} \tag{7}$$

$$\Psi_f = \begin{bmatrix} Y_f \\ U_f \end{bmatrix} \tag{8}$$

By recursive substitution into the state space model equations Eqs. (1,2), it is straightforward to show:

$$Y_f = \Gamma_{i_H} X_f + \Phi^d_{i_H} U_f \tag{9}$$

$$Y_p = \Gamma_{i_H} X_p + \Phi^d_{i_H} U_p \tag{10}$$

$$X_f = A^{i_H} X_p + \Delta^d_{i_H} U_p \tag{11}$$

where:

$$\Gamma_{i_H} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i_H-1} \end{bmatrix} \tag{12}$$

$$\Phi^d_{i_H} = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CA^{i_H-2}B & CA^{i_H-3}B & CA^{i_H-4}B & \dots & D \end{bmatrix} \tag{13}$$

$$\Delta^d_{i_H} = \begin{bmatrix} A^{i_H-1}B & A^{i_H-2}B & \dots & AB & B \end{bmatrix} \tag{14}$$

Solving for X_f in Eq. 9 yields

$$X_f = \begin{bmatrix} \Gamma^\dagger_{i_H} & -\Gamma^\dagger_{i_H}\Phi_{i_H} \end{bmatrix} \begin{bmatrix} Y_f \\ U_f \end{bmatrix} \tag{15}$$

where $\Gamma^\dagger_{i_H}$ denotes the pseudo-inverse of Γ_i. It can be concluded from Eq. 15 that the row space of X_f is comprised within the row space of $\begin{bmatrix} Y_f^T & U_f^T \end{bmatrix}^T$. Similarly, X_p can be calculated from Eq. 10 and by substituting into Eq. 11, we can write:

$$X_f = \begin{bmatrix} A^{i_H}\Gamma^\dagger_{i_H} & \Delta_{i_H} - A^{i_H}\Gamma^\dagger_{i_H}\Phi_{i_H} \end{bmatrix} \begin{bmatrix} Y_p \\ U_p \end{bmatrix} \tag{16}$$

From equation 16 it can be concluded that the row space of X_f is contained within the row space of $\begin{bmatrix} Y_p^T & U_p^T \end{bmatrix}^T$. Then, X_f can be calculated from the intersection between the past and future data:

$$\text{span}\left(X_f\right) \coloneqq \text{row space}\left(\begin{bmatrix} Y_f \\ U_f \end{bmatrix}\right) \cap \text{row space}\left(\begin{bmatrix} Y_p \\ U_p \end{bmatrix}\right) \tag{17}$$

In,[23] a computationally efficient method was proposed to estimate state sequence using singular value decomposition (SVD) method. Then, LTI model parameters can be calculated using the state sequence.[24]

The model matrices can be estimated using least squares:

$$\begin{bmatrix} X_f \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_p \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} W_{i|i} \\ V_{i|i} \end{bmatrix}. \tag{18}$$

The system matrices can be calculated as follows:

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \begin{bmatrix} X_f \\ Y_{i|i} \end{bmatrix} \begin{bmatrix} X_p \\ U_{i|i} \end{bmatrix}^{\dagger}. \tag{19}$$

With the proposed approach, process and measurement noise Hankel matrices can be

calculated as the residual of the least square solution of Equation (18):

$$\begin{bmatrix} \hat{W}_{i|i} \\ \hat{V}_{i|i} \end{bmatrix} = \begin{bmatrix} X_f \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} X_p \\ U_{i|i} \end{bmatrix}. \tag{20}$$

Then, the covariances of plant noises can be estimated as follows:

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = E(\begin{bmatrix} \hat{W}_{i|i} \\ \hat{V}_{i|i} \end{bmatrix} \begin{bmatrix} \hat{W}_{i|i}^T & \hat{V}_{i|i}^T \end{bmatrix}). \tag{21}$$

Remark 1 *The information on the number of states to be used for the model (which really is the only model structure choice) is determined from the availability of the data itself. Specifically, at the singular value decomposition step, the choice of the number of states directly yields the required level of approximation, and this can be utilized to pick the number of states (a good rule of thumb is to pick enough states to achieve 90-95 % accuracy). Note also that for cases with noise or disturbance in the plant, open-loop identification methods are theoretically biased for system identification, and closed-loop identification must be used. Note that in this work we look into the deterministic problem, so we were able to use an open-loop method. Also, the re-identification step can use any kind of identification method, as an instance the closed-loop identification method utilized in.*[16,25]

2.2.1 Formation of batch data Hankel matrices

For batch processes training data are taken from different batches with different duration to ensure data richness. Therefore in constructing data-Hankel matrices, input and output data can not be handled as one block of data. In order to handle this issue, in,[24] it was proposed to construct data-Hankel matrix for each batch b with input variable as the Hankel sub-matrix as:

$$U_p^{(b)} = U_{1|i_H}^{(b)} = \begin{bmatrix} u_1^{(b)} & u_2^{(b)} & \dots & u_{j^{(b)}}^{(b)} \\ \vdots & \vdots & \vdots & \vdots \\ u_{i_H}^{(b)} & u_{i_H+1}^{(b)} & \dots & u_{i_H+j^{(b)}-1}^{(b)} \end{bmatrix} \tag{22}$$

The overall data-Hankel matrix is constructed by concatenating these sub-Hankel matrices horizontally:

$$U_p = \begin{bmatrix} U_p^{(1)} & U_p^{(2)} & \dots & U_p^{(B)} \end{bmatrix} \tag{23}$$

where B indicates the total number of the batches. By utilizing the described data-Hankel matrix, the illustrated subspace identification can compute the LTI model matrices to be used in MPC.

2.3 Model predictive control

A representative model predictive control formulation for trajectory control of batch processes is as follows:

$$\min_{\tilde{u}_k,\ldots,\tilde{u}_{n_t}} \sum_{j=k}^{n_t} ||\tilde{y}_{k+j} - \tilde{y}^{\mathrm{SP}}_{k+j}||^2_{Q_y} + ||\tilde{u}^2_{j-1} - \tilde{u}^2_{j-2}||^2_{R_{du}},$$

subject to:

$$\begin{aligned} \tilde{x}_{k+1} &= A\tilde{x}_k + B\tilde{u}_k, \\ \tilde{y}_k &= C\tilde{x}_k + D\tilde{u}_k, \\ \tilde{u} &\in \mathcal{U}, \qquad \tilde{x}(k) = \hat{x}_l, \end{aligned} \tag{24}$$

where n_t denotes the end of the batch, $\tilde{y}^{\mathrm{SP}}_k$ is the desired output (desired trajectory), and, $\tilde{y}_k$ and $\tilde{u}_k$ are the predicted output trajectory and input at the time $k\Delta t$. $Q_y \in R^{n\times n}$ and R_{du} are positive definite, and, positive semi-definite matrices respectively, and they are chosen so the nominal closed-loop system is stable.[26] $\tilde{x}$ and $\hat{x}$ are predicted and the estimate of the subspace state, obtained using an appropriate state estimator. For illustrative purposes (and for the simulations in this paper), a Kalman filter is employed for state estimation. The

state estimator has the following form:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

$$P_k^- = AP_{k-1}A^T + Q$$

$$K_k = P_k^- C^T \left(CP_k^- C^T + R\right)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k \left(y_k - C\hat{x}_k^-\right)$$

$$P_k = \left(I - K_k C\right) P_k^-$$

where $\hat{x}_k^-$ and P_k^- indicate state and covariance matrix prediction at sample time k. Q and R are state and output covariance matrices, K_k is the Kalman filter gain at sample time k and I i denotes the identity matrix.

Remark 2 *Note that there exist different MPC formulations for batch processes, including those where end point constraints are utilized, and those that achieve quality control. A generic MPC formulation is employed to simply illustrate the proposed MPC with re-identification approach. By the same token, the control design could be implemented coupled with other estimation approaches such as the moving horizon estimation, and the Kalman filter is simply being used for illustrative purposes.*

2.4 Model predictive control with re-identification for continuous operation

We next review a recently proposed monitoring based MPC formulation.[16] In this formulation, at a sample time k, the updated state estimate archived at sample time $k-h$, i.e., $\hat{x}_{k-h}$ and the input sequence $u_{k-h}, u_{k-h+1}, \ldots, u_{k-1}$, where h denotes the monitoring horizon for the model are utilized to compute the 'predicted' behavior of the plant $(\hat{y}_{k-h}, \hat{y}_{k-h+1}, \ldots, \hat{y}_{k-1})$. Since the true plant output is measured and stored for this period, this prediction is used to evaluate model prediction performance. A schematic presentation of this approach is presented in Figure (1). In the schematic in Figure 1, the predicted output

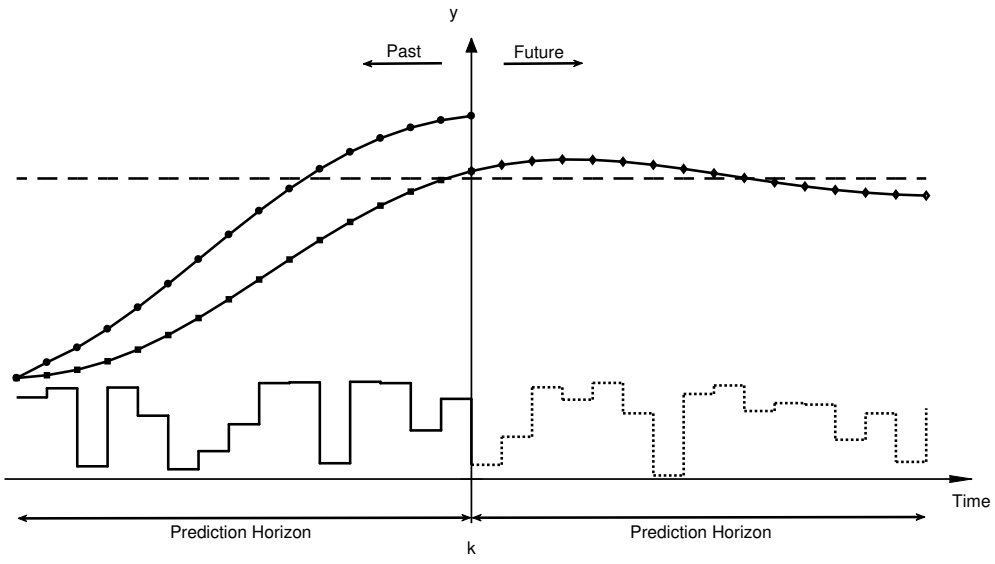

Figure 1: A schematic depicting an MPC implementation with re-identification for continuous operation −■− denotes the measured output, −●− denotes the past predicted output, −◆− denotes the Future Predicted Output, −− denotes Setpoint, − denotes Past Input, .. denotes Future Predicted Input)

trajectory is calculated utilizing the state space LTI model with estimated state vector at the sample time $k-h$ (which is stored at each sample time for the purpose of model monitoring) and the implemented input from $k-h$ to $k-1$. The difference between measured and predicted output is ascribed to plant-model mismatch. The future computed input trajectory and the the future predicted output, are calculated by the MPC at the current sample time. Finally, note that the model monitoring does not need solving an optimization problem, but only requires the integration of the identified model with known initial condition and input trajectory.

From a model monitoring perspective, the key difference between continuous operation and batch processes is appropriately accounting for the batch nature of process data. In general, a direct application of the model monitoring approach for continuous operation will lead to erroneous conclusions for batch operation. The model performance evaluation and re-identification approach for batch processes is presented next.

3 Model monitoring and re-identification based model predictive batch process control

In this section, we first present a model monitoring approach for batch processes. Then, a re-identification approach that enables greater emphasis on recent data is presented.

3.1 Model performance index

In,[16] two model prediction perforce indexes were proposed, end point prediction error (EPPE) and mean prediction error (MPE). In this work the MPE index is adapted for quantifying model mismatch in batch processes. The MPE index is calculated as follows:

$$MPE_k = \frac{1}{h} \sum_{j=k-h}^{k-1} (|y_j - \bar{y}_j|) \tag{25}$$

where $MPE_k \in R^{n_y}$ is the mean prediction error, and is calculated as the mean value of prediction error along the model monitoring horizon at sample time k and, $\bar{y}$ is the corresponding output prediction. As opposed to performance monitoring indexes, the proposed index directly tests the predictive capability of the model. One of the key differences in batch implementation is the determination of the threshold. One proposed approach for threshold determination is as follows:

$$MPE_{k,i} = max\{MPE_{k,i}^{(1)}, \ldots, MPE_{k,i}^{(B)}\}, \quad \text{For } i = 1, \ldots, n_y \tag{26}$$

Thus, the threshold at each sample time is chosen as the maximum value of the index among all the batches, at the same time in the training batches.

Remark 3 *The time varying index renders the ability to set time varying threshold. The*

method, however, needs to be carefully adapted when dealing with batches of different duration. One possibility is the use of an alignment variable. Alternatively, the maximum value of the index over the training batches, could be used. Another possibility is one where the error threshold is computed based on the proximity in state space in the training batches. The explicit illustration of these ideas remains outside the scope of the present work.

Remark 4 *Note that the key contribution of the present manuscript is not the model identification algorithm, but the monitoring and re-identification approach. Thus, while the use of multiple linear models as the base models has not been explicitly shown, the monitoring and re-identification idea could very well be adapted to the situation where multiple linear models are being utilized. In such a scenario, the monitoring approach would indicate the requirement of a new 'local' linear model (if the existing multiple linear models do not sufficiently predict the process dynamics) if the process evolution encounters a new region in state space where the dynamics are significantly different from the original identification data. The simulations do consider disturbances 'natural' to the batch process, such as variation of the initial conditions. Future work will utilize other processes where a time varying disturbance would be natural to the process (such as changing external air temperature).*

3.2 Model monitoring

Beyond the definition of the threshold, the other key difference in the batch process monitoring is the requirement to wait for the state estimator to converge before starting the process monitoring. The schematic presentation of MPC with re-identification for batch processes is presented in Figure (2). In order to evaluate model prediction performance at each sample

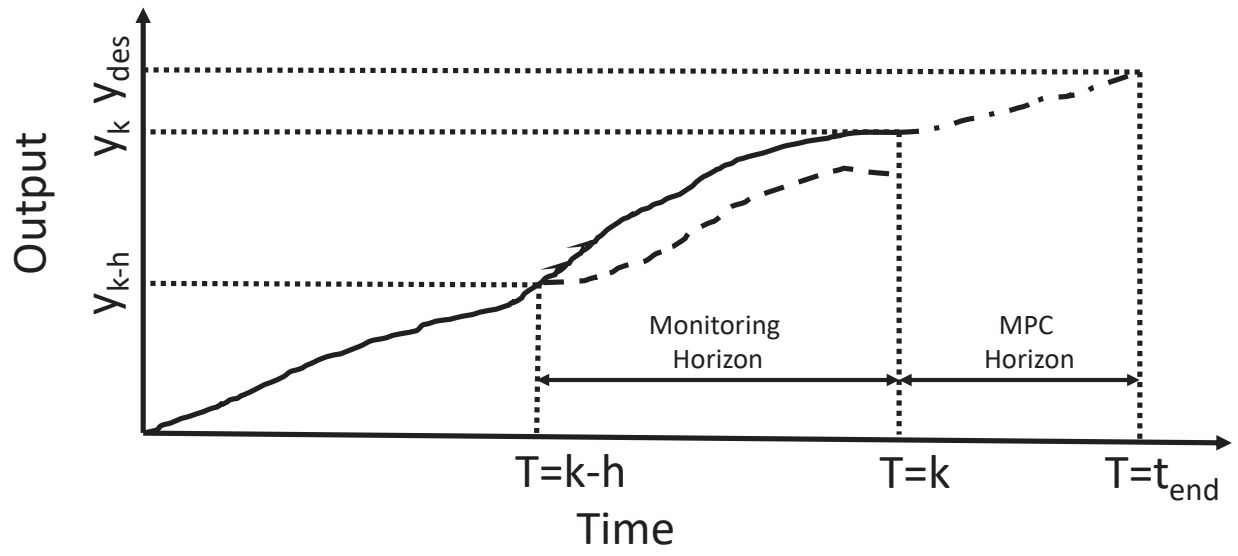

Figure 2: MPC with re-identification for batch systems ([Measured Output:continuous line], [Past Predicted Output: dashed line], [Future Predicted Output:dashed-dotted line])

time l, the estimated state vector x_{l-h} at sample time $l-h$ and the input trajectory from

u_{l-h} to u_{l-1} are used to calculate the past output prediction as follows:

$$\begin{gathered} \bar{y}_j = C\bar{x}_j + Du_j \\ \bar{x}_{j+1} = A\bar{x}_j + Bu_j \\ \text{where } j = l-h, \ldots, l-1 \\ \bar{x}_l = \hat{x}_{l-h} \end{gathered} \tag{27}$$

where $\bar{y}$ and $\bar{x}$ are predicted output and state. Model monitoring, however, is only initiated after a certain time into the batch (denoted by sampling index k^*). One of the criterion for determining k^* is that the state estimator achieve a required convergence. The state estimator is said to have converged at sample time k if:

$$|y_{i,k} - \hat{y}_k| < \epsilon_i^* \quad \text{For } i = 1, \ldots, n_y \tag{28}$$

where ϵ^* denotes the vector of acceptable output prediction error, and k^o represents the smallest sample time at which condition in equation 28 is satisfied.

$$k^o = inf\{k||y_{i,k} - \hat{y}_k| < \epsilon_i^* \quad \text{For } i = 1, \ldots, n_y\} \tag{29}$$

Furthermore, in order to have enough data to fill a column in Hankel matrices, a certain number of samples need to have been collected(i_H in Equation (22)). Therefor, k^* is determined

as follows:

$$k^* = h + max\{k^o, i_H\} \tag{30}$$

Remark 5 *The requirement to wait a certain number of samples before initiating model monitoring is consistent with model predictive control implementations in batch processes. In particular, the subspace model based predictive control designs are set to activate only after the state estimates have converged, providing reliable state estimates, and predictive capability. The model monitoring needs to wait possibly longer for two reasons. The first one being the requirement for the state estimates to have converged, and a reasonable amount of time have to have passed after the convergence of the estimates to test model predictions. The second requirement ensures that sufficient data has been collected to re-identify, should the re-identification be triggered.*

Remark 6 *The monitoring horizon is chosen to be more than the controller horizon. This is to be consistent with the impact that the model has- that of deciding on the control action through the prediction. Using an unnecessarily long monitoring horizon would simply lead to a growth in error, and in turn require larger thresholds. A monitoring horizon just larger than the control horizon is sufficient to detect and fix the incorrect impact the model would have on the closed-loop behavior.*

3.3 Re-identification for batch systems

At any sampling instance where the index breaches its threshold, a re-identification is carried out. In the re-identification step, the current batch measurements are concatenated with past training data horizontally to create Hankel matrices. More importantly, to emphasize more the recent dynamics, the measurements from the current batch are repeated in the Hankel matrix as follows:

$$U_p = \begin{bmatrix} U_p^{(1)} & U_p^{(2)} & \dots & U_p^{(B)} & U_p^{(B)} & \dots & U_p^{(B)} \end{bmatrix} \tag{31}$$

The number of repetitions is a tuning parameter and can be informed by the number of batches in initial training data, in a way to make sure that recent data is dominant. For instance, in the present work, the number of repetition are chosen so the recent data amounts to twenty percent of the total data in the augmented Hankel matrix. Note that, repeating the current batch multiple times is the easiest way of incorporating more weight on the current batch, without have to change the regression problem.

Remark 7 *Note that in the proposed method the computation of the past output prediction is for a purpose different from the moving horizon estimation (MHE) method, which estimates past states using an optimization problem. The MHE approach can not be utilized to check*

for model validity, because in the MHE approach, the state values at the past point in time are computed (the state values are one of the decision variables in the optimization problem), and done in a way that minimizes the error between the predictions and observed variables. While it is an excellent state estimation technique, the computation of the state estimate defeats the model monitoring objective. In contrast, in the proposed approach, the state estimates generated by the state estimator is used to focus on the model validity aspect. That said, the MHE can be readily utilized instead of the Kalman filter for the purpose of state estimation.

Remark 8 *The key advantage of the use of the pseudo-Hankel matrices formed by the concatenation of batch data is that it allows the ability to 'naturally' emphasize the recent data by simply creating multiple instances of the recent data. Such a natural concatenation does not readily apply in the continuous time setting. The other advantage of concatenating horizontally is that the data-Hankel matrices can consist of batches with different durations, without the need of the existence of an appropriate alignment variable.*

At each sample time model prediction is evaluated and the LTI model used by MPC is updated only if the initial LTI model prediction is poor (i.e., the threshold is breached) and the new model has improved prediction.[16] After model identification, the previous measured outputs of the current batch are utilized with the new model to calculate current state with the new model using Kalman filter, both for model prediction checking, and to initialize the

MPC (see the algorithm in the next section for a better understanding).

3.4 Model monitoring and re-identification based MPC for batch processes

In this section, we briefly describe through an algorithm the implementation of the re-identification based MPC. The algorithm describes the sequence of steps after a model (denoted by M_1) has been identified using training data, and the thresholds for model monitoring determined.

1. Initialize the Kalman filter using the existing model and a guess for the initial state (see Remark 9 below for further discussion on this point), and run the process under open-loop/PID controller up until the outputs converge.

2. After the outputs converge, engage the MPC, and check conditions for model prediction testing.

3. Upon satisfaction of the model prediction testing criteria, initiate model monitoring.

4. If at any time during the batch, the prediction monitoring threshold is breached successively more than $b*$ times, where $b*$ is a user defined parameter

   (a) Re-identify a model (denoted by M_2) by concatenating the training data with

repeats of the current batch data.

(b) Using a guess for an initial state from the start of the batch, run the Kalman filter with the new model, up-until h time steps back from the current time and evaluate the predictive capability of the new model over h sample times (see Remark 10 on why this is needed).

(c) If the new model has improved prediction, then continue the Kalman filtering to determine the state estimate at the current time, and using this state estimate, compute the control action under the MPC.

5. Continue monitoring, checking step 4, until batch termination.

Remark 9 *In general, the initial state of a new batch remains an unknown quantity (only the outputs are directly measured). That said, starting from an initial guess that might be closer to the 'true' value of the subspace states for that particular batch would certainly help with the convergence of the state estimation scheme and favor early engagement of the MPC. One way to determine an initial guess that will possibly be closer to the true state values invokes the assumptions that most batches (with their initial conditions) are designed to be sufficiently close to each other. Thus, in this work, the initial state estimate guess (that is utilized to initialize the Kalman filter) is determined by averaging the initial state value (computed at the time of model identification) over the training batches.*

Remark 10 *The parameter $b*$ serves to prevent repetitive re-identification. Thus, re-identification is triggered only after the threshold is breached a certain number of times (with values typically being about half the monitoring horizon). Furthermore, before engaging the new model, one needs to determine if the new model predicts better than the previous model for the current batch. To do this, one must be able to 'predict' using the new model over the past h time steps. To perform this prediction, and to have a fair comparison with the old model, the new model must be provided state estimates that are consistent with the new model. To achieve this, step 4b requires the appropriate estimation of the states for the new model, and using those state estimates with the new model to test the predictive capability.*

4 Application to the Electric Arc Furnace

In this section, first an electric arc furnace (EAF) process is described, then the efficiency of the identification is demonstrated. Finally, the identified state space LTI model is utilized in the proposed model predictive control with re-identification approach, and simulation results on a test bed indicates improvement in closed-loop behavior.

4.1 Electric arc furnace process description

Electric arc furnace (EAF) process is used to produce steel from recycling scrap and direct reduced iron. The EAF process is a batch process, which starts with scrap metal being loaded inside the furnace. The duration of each batch is about one to two hours. EAF utilizes a high intensity electric arc, in order to melt the scrap metal. The electric arc is usually the largest energy consumer in the EAF process. After most of the metal has been melted, in order to create iron oxide and carbon monoxide, raw carbon and oxygen gas are injected into the molten steel. The batch is terminated when the desired steel composition and temperature are obtained (detailed explanation of EAF process and modeling details and formulations can be found in,[24] and are omitted here for brevity). A list of the process output variables are given in Table 1, and the inputs are listed in Table 2. The outputs y_1, y_5, y_6 and y_7 are controlled outputs. The controller ensures that the controlled outputs follow their set-point trajectory.

4.2 Electric arc furnace model identification

As a test bed, a first principles EAF[27] process model is utilized. 40 normal operation batches of varying durations between 60 to70 sample times are assumed to be available for training. Before proceeding with model identification, full row rank of input Hankel matrix

Table 1: List of Output Variables of the EAF Process

Variable Name	Variable	Description	Units
y_1	T	Temperature of Molten Steel	K
y_2	x_{Fe}	Mass Fraction Iron in Molten Steel	kg/kg
y_3	x_{C}	Mass Fraction Carbon in Molten Steel	kg/kg
y_4	x_{Slag}	Mass Fraction Lime/Dolime in Slag	kg/kg
y_5	x_{FeO}	Mass Fraction Iron Oxide in Slag	kg/kg
y_6	$x_{\mathrm{SiO_2}}$	Mass Fraction Silicon Dioxide in Slag	kg/kg
y_7	P	Relative Pressure	Pa
y_8	x_{CO}	Mass Fraction Carbon Monoxide in Gas	kg/kg
y_9	$x_{\mathrm{CO_2}}$	Mass Fraction Carbon Dioxide in Gas	kg/kg
y_{10}	$x_{\mathrm{N_2}}$	Mass Fraction Nitrogen in Gas	kg/kg

Table 2: List of Manipulated Variables for the EAF Process

Variable Name	Variable	Description	Units
u_1	m_{gas}	Off-gas Turbine Flow	kg/s
u_2	$m_{\mathrm{O_2}}$	Oxygen Lanced	kg/s
u_3	m_{DRI}	DRI Additions	kg/s
u_4	m_{Slag}	Slag Additions	kg/s
u_5	E	Electric Arc Power	kW
u_6	m_{C}	Carbon Injected	kg/s

was ascertained. Subsequently, an LTI model with 12 states is identified.

For validation, a different batch of data was utilized. The model validation results are presented in Figures (3 & 4). Since the initial state of LTI model is not available for new batch, Kalman filter is utilized for initial sample times and after output convergence, open-loop prediction is used for model validation. At a time 30 minutes into the batch, the output of Kalman filter converge to the plant output, and the identified LTI model in open-loop (without state update), along with the known input trajectory, is utilized for output prediction with the remainder of the batch. The results indicate that after convergence of

model states, the model is capable of predicting the process behavior reasonably well, and is suitable for an MPC implementation.

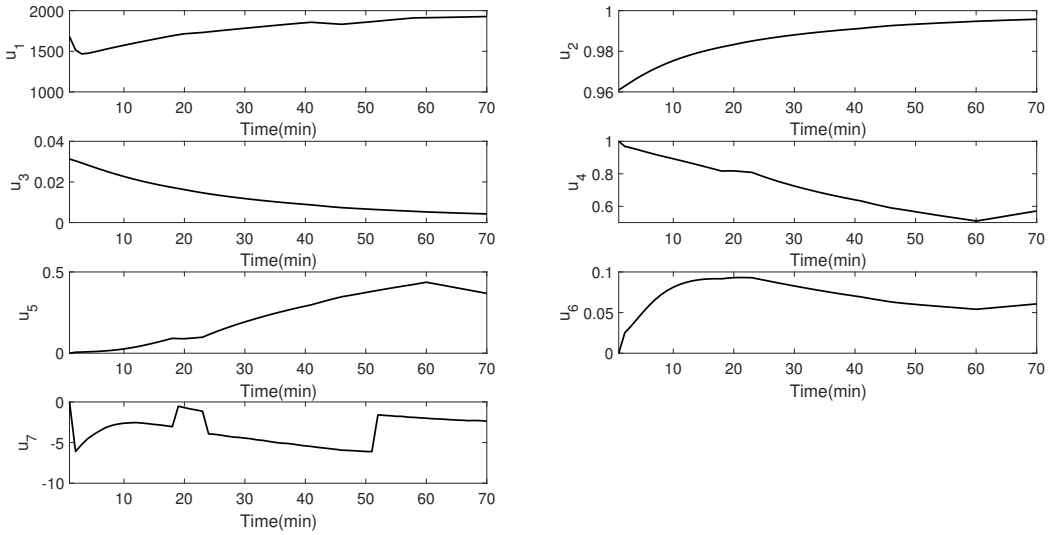

Figure 3: Input profiles for the validation batch

4.3 Model predictive control of the electric arc furnace

The proposed MPC with monitoring and identification was implemented on new batches (see Table 3 for the controller parameters). For the purpose of model monitoring, the parameters h and $b*$ were chosen as 10, and 6, respectively. The simulation results show comparison of three MPC formulations. The first is the standard MPC without re-identification, the second is with re-identification, where, the recent data is simply added to the training data and a model is identified. The third implementation is where the model is identified using ten repetitions of the data from the current batch. Following the algorithm outlined in

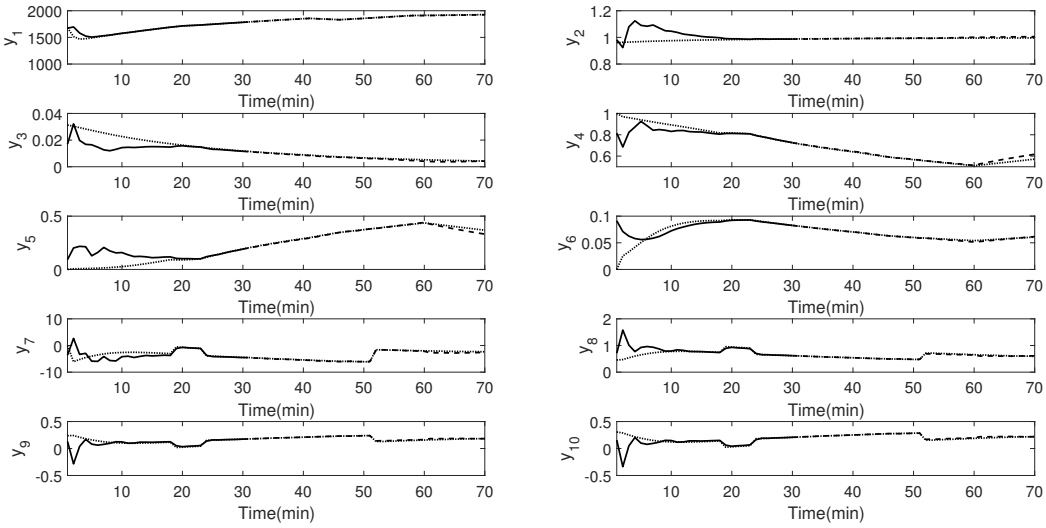

Figure 4: Validation batch: measured outputs (continuous line) and predicted output (dashed line)

Section 3.4, re-identification occurs at sample times $28, 34,\ 40,\ 46$ and 52, for both the MPC implementations with re-identification. The multiple trigger re-identification is expected as the batch process moves through significantly different dynamics.

As can be seen from Figures (5-7). the proposed MPC that emphasizes the recent data more is the one that yields a trajectory closest to the reference trajectory (designed for four output variables). The superior performance was also ascertained quantitatively, using the root mean square error of all the controlled variables, and presented in Table (4). More importantly, the improved predictive ability of the re-identified models can be seen from the lower values of the monitoring index in Figure 7.

Remark 11 *In the proposed method at each sample time k, state estimation x_{k-h} and past*

input $u_{k-h}, \ldots, u_{k-1}$ is utilized to calculate output trajectory, $y_{k-h}, \ldots, y_{k-1}$. None of these variables (past input trajectory and state estimates) need to be computed in the proposed framework but simply need to be read from the data historian.

Table 3: Controller parameters

Variable	Value
Q_y	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
R_{du}	0
Monitoring Horizon	10
k^*	28
ϵ^*	0.001

Table 4: Model validation results

Variable Name	RMSE_{MPC}	RMSE_1	RMSE_{10}	Unit
y_1	159.49	137.53	137.52	K
y_5	0.0814	0.0685	0.0682	kg/kg
y_6	0.0121	0.0101	0.0101	kg/kg
y_7	0.7364	0.6624	0.6319	Pa

5 Conclusions

An MPC with re-identification framework is presented for batch processes that enables model prediction performance monitoring and adapting the model used in the MPC using both initial training data and data from the current batch. The MPC with re-identification method is demonstrated and compared against traditional MPC using an EAF process simulation

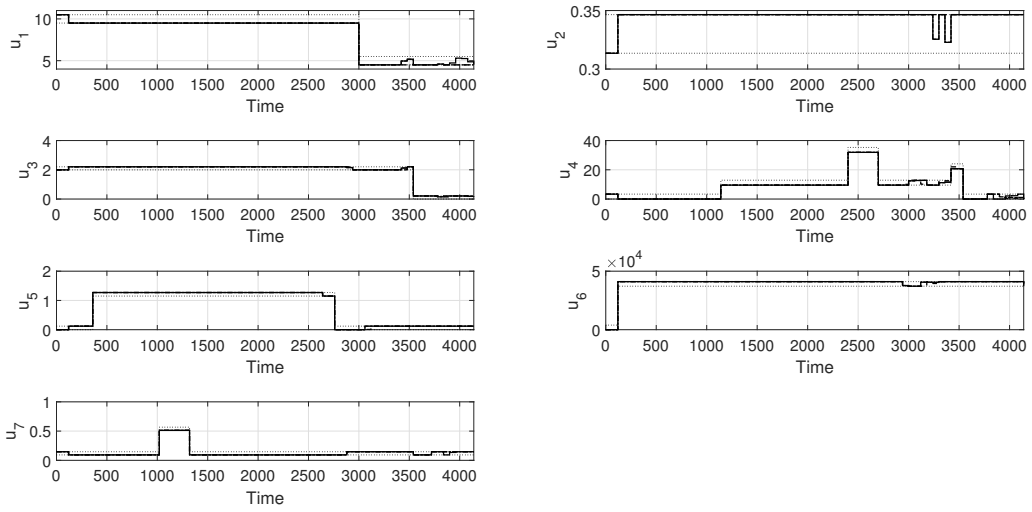

Figure 5: Closed-loop profiles of the input variables obtained from the proposed MPC and nominal MPC (Nominal MPC: dashed line, MPC with re-identification with one repetition : continuous line, MPC with re-identification with ten repetitions: dash-dotted line and lower and upper bounds on the inputs: dotted lines)

as a test bed. The simulations illustrate the ability of the proposed method to improve closed-loop performance.

6 Acknowledgments

Financial support from the McMaster Advanced Control Consortium (MACC) is gratefully acknowledged.

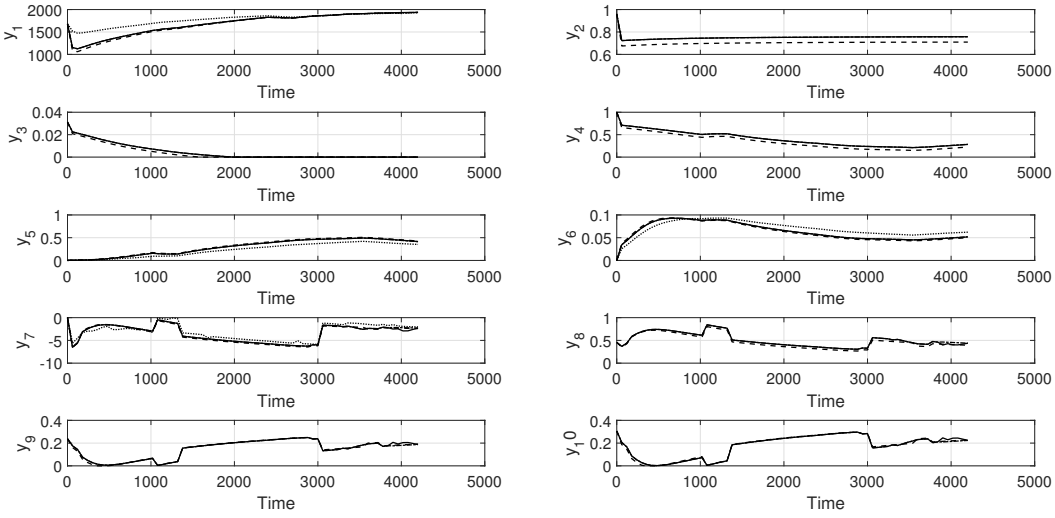

Figure 6: Comparison of the trajectories output variables obtained from the proposed MPC with re-identification and nominal MPC (Nominal MPC: dashed line, MPC with re-identification with one repetition : continuous line, MPC with re-identification with ten repetitions : dash-dotted line) and set-point for the controlled outputs: dotted line

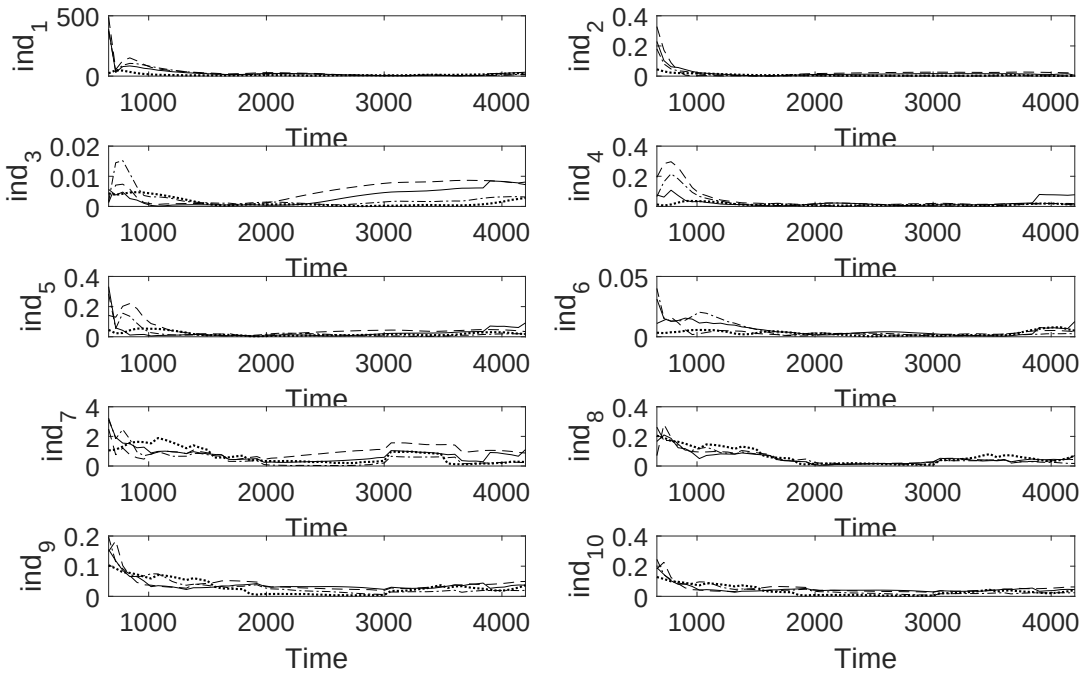

Figure 7: Comparison of the index trajectories obtained from the proposed MPC with re-identification and nominal MPC (Nominal MPC: dashed line, MPC with re-identification with one repetition : continuous line, MPC with re-identification with ten repetitions : dash-dotted line and threshold: dotted lines)

References

(1) Flores-Cerrillo, J.; MacGregor, J. F. Latent variable MPC for trajectory tracking in batch processes. *Journal of process control* **2005**, *15*, 651–663.

(2) Flores-Cerrillo, J.; MacGregor, J. F. Control of batch product quality by trajectory manipulation using latent variable models. *Journal of Process Control* **2004**, *14*, 539–553.

(3) Chin, I. S.; Lee, K. S.; Lee, J. H. A technique for integrated quality control, profile control, and constraint handling for batch processes. *Industrial & engineering chemistry research* **2000**, *39*, 693–705.

(4) Lee, J. H.; Lee, K. S. Iterative learning control applied to batch processes: An overview. *Control Engineering Practice* **2007**, *15*, 1306–1318.

(5) Bonvin, D. Control and optimization of batch processes. *Encyclopedia of Systems and Control* **2015**, 133–138.

(6) Rawlings, J. B.; Mayne, D. Q. *Model predictive control: Theory and design*; Nob Hill Pub., 2009.

(7) Mayne, D. Q.; Raković, S.; Findeisen, R.; Allgöwer, F. Robust output feedback model predictive control of constrained linear systems. *Automatica* **2006**, *42*, 1217–1222.

(8) Mahmood, M.; Gandhi, R.; Mhaskar, P. Safe-parking of nonlinear process systems: Handling uncertainty and unavailability of measurements. *Chemical Engineering Science* **2008**, *63*, 5434–5446.

(9) Pannocchia, G.; Rawlings, J. B. Disturbance models for offset-free model-predictive control. *AIChE journal* **2003**, *49*, 426–437.

(10) Wallace, M.; Pon Kumar, S. S.; Mhaskar, P. Offset-free model predictive control with explicit performance specification. *Industrial & Engineering Chemistry Research* **2016**, *55*, 995–1003.

(11) Narasingam, A.; Siddhamshetty, P.; Kwon, J. S.-I. Handling Spatial Heterogeneity in Reservoir Parameters Using Proper Orthogonal Decomposition Based Ensemble Kalman Filter for Model-Based Feedback Control of Hydraulic Fracturing. *Industrial & Engineering Chemistry Research* **2018**, *57*, 3977–3989.

(12) Narasingam, A.; Kwon, J. S.-I. Development of local dynamic mode decomposition with control: Application to model predictive control of hydraulic fracturing. *Computers & Chemical Engineering* **2017**, *106*, 501–511.

(13) Huang, B.; Shah, S. L. *Performance assessment of control loops: theory and applications*; Springer Science & Business Media, 2012.

(14) Shah, S. L.; Patwardhan, R.; Huang, B. Multivariate controller performance analysis: methods, applications and challenges. AICHE Symposium Series. 2002; pp 190–207.

(15) Chilin, D.; Liu, J.; Davis, J. F.; Christofides, P. D. Data-based monitoring and reconfiguration of a distributed model predictive control system. *International Journal of Robust and Nonlinear Control* **2012**, *22*, 68–88.

(16) Kheradmandi, M.; Mhaskar, P. Model predictive control with closed-loop re-identification. *Computers & Chemical Engineering* **2018**, *109*, 249–260.

(17) Genceli, H.; Nikolaou, M. New approach to constrained predictive control with simultaneous model identification. *AIChE journal* **1996**, *42*, 2857–2868.

(18) Potts, A. S.; Romano, R. A.; Garcia, C. Improving performance and stability of MPC relevant identification methods. *Control Engineering Practice* **2014**, *22*, 20–33.

(19) Heirung, T. A. N.; Ydstie, B. E.; Foss, B. Towards dual MPC. *IFAC Proceedings Volumes* **2012**, *45*, 502–507.

(20) Van Overschee, P.; De Moor, B. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica* **1994**, *30*, 75–93.

(21) Huang, B.; Kadali, R. *Dynamic modeling, predictive control and performance monitoring: a data-driven subspace approach*; Springer, 2008.

(22) Zhao, Y.; Qin, S. J. Subspace identification with non-steady Kalman filter parameterization. *Journal of Process Control* **2014**, *24*, 1337–1345.

(23) Moonen, M.; De Moor, B.; Vandenberghe, L.; Vandewalle, J. On-and off-line identification of linear state-space models. *International Journal of Control* **1989**, *49*, 219–232.

(24) Rashid, M. M.; Mhaskar, P.; Swartz, C. L. Handling multi-rate and missing data in variable duration economic model predictive control of batch processes. *AIChE Journal* **2017**, *63*, 2705–2718.

(25) Kheradmandi, M.; Mhaskar, P. Data Driven Economic Model Predictive Control. *Mathematics* **2018**, *6*, 51.

(26) Kheradmandi, M.; Mhaskar, P. Prescribing Closed-Loop Behavior Using Nonlinear Model Predictive Control. *Industrial & Engineering Chemistry Research* **2017**, *56*, 15083–15093.

(27) Bekker, G. J.; Craig, I. K.; Pistorius, P. C. Model predictive control of an electric arc furnace off-gas process. *Contr. Eng. Pract.* **2000**, *8*, 445 – 455.

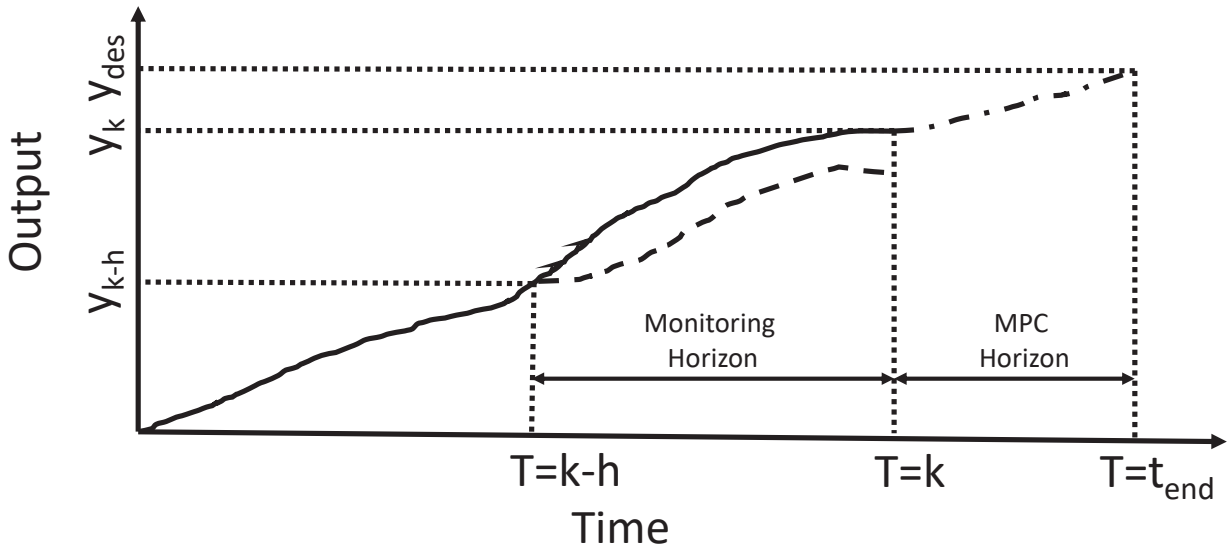

Figure 8: For Table of Contents Only