ADVANCES IN MODEL PREDICTIVE CONTROL

ADVANCES IN MODEL PREDICTIVE CONTROL

by

MASOUD KHERADMANDI, M.SC.

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

DOCTOR OF PHILOSOPHY (2018) McMaster University
(Chemical Engineering) Hamilton, Ontario, Canada

Title: Advances in Model Predictive Control

Author: Masoud Kheradmandi, M.Sc. (Chemical Engineering)
McMaster University, Hamilton, Ontario, Canada

Supervisor: Dr. Prashant Mhaskar

Number of pages: xx, 145

In the name of Allah, the Most Beneficent, the Most Merciful.

ABSTRACT

In this thesis I propose methods and strategies for the design of advanced model predictive control designs. The contributions are in the areas of data-driven model based MPC, model monitoring and explicit incorporation of closed-loop response considerations in the MPC, while handling issues such as plant-model mismatch, constraints and uncertainty.

In the initial phase of this research, I address the problem of handling plant-model mismatch by designing a subspace identification based MPC framework that includes model monitoring and closed-loop identification components. In contrast to performance monitoring based approaches, the validity of the underlying model is monitored by proposing two indexes that compare model predictions with measured past output. In the event that the model monitoring threshold is breached, a new model is identified using an adapted closed-loop subspace identification method. To retain the knowledge of the nominal system dynamics, the proposed approach uses the past training data and current input, output and set-point as the training data for re-identification. A model validity mechanism then checks if the new model predictions are better than the existing model, and if they are, then the new model is utilized within the MPC.

Next, the proposed MPC with re-identification method is extended to batch processes. To this end, I first utilize a subspace-based model identification approach for batch processes to be used in model predictive control. A model performance index is developed for batch process, then in the case of poor prediction, re-identification is triggered to identify a new model. In order to emphasize

on the recent batch data, the identification is developed in order to increase the contribution of the current data. In another direction, the stability of data driven predictive control is addressed. To this end, first, a data-driven Lyapunov-based MPC is designed, and shown to be capable of stabilizing a system at an unstable equilibrium point. The data driven Lyapunov-based MPC utilizes a linear time invariant (LTI) model cognizant of the fact that the training data, owing to the unstable nature of the equilibrium point, has to be obtained from closed-loop operation or experiments. Simulation results are first presented demonstrating closed-loop stability under the proposed data-driven Lyapunov-based MPC. The underlying data-driven model is then utilized as the basis to design an economic MPC.

Finally, I address the problem of control of nonlinear systems to deliver a prescribed closed-loop behavior. In particular, the framework allows for the practitioner to first specify the nature and specifics of the desired closed-loop behavior (e.g., first order with smallest time constant, second order with no more than a certain percentage overshoot, etc.). An optimization based formulation then computes the control action to deliver the best attainable closed loop behavior. To decouple the problems of determining the best attainable behavior and tracking it as closely as possible, the optimization problem is posed and solved in two tiers. In the first tier, the focus is on determining the best closed-loop behavior attainable, subject to stability and tracking constraints. In the second tier, the inputs are tweaked to possibly improve the tracking of the optimal output trajectories given by the first tier. The effectiveness of all of the proposed methods are illustrated through simulations on nonlinear systems.

ACKNOWLEDGMENTS

All the praises and thanks are due to God, the Lord of the Worlds.

First and foremost, I would like to thank my parents, Mrs Fatemeh Nilforosh and Mr Majid Kheradmandi, for being the best teachers in my life. I would also like to thank my sister, Mahsa Kheradmandi. This work would not have been possible without the unconditional love and support of my family. This dissertation is dedicated to my parents, for all their prayers, sacrifices, love, and support.

I am extremely grateful to my advisor, Dr. Prashant Mhaskar, and thank him for all his patience and guidance. It has been a great honor to work with him, and I am greatly appreciative of his leadership, support and encouragement. He has instilled in me an appreciation for meaningful research. I would also like to thank the members of my committee, Dr. Thomas Adams and Dr. Fengjun Yan, for their insights, advice, and time.

There are also many friends who deserve my appreciation and thanks, both in Hamilton and beyond. Thanks also goes to Farbod Kohan and Mudassir Rashid for their numerous valuable and insightful conversations and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Background and Motivation

The operation of chemical plants faces numerous challenges such as inherent nonlinearity, complex variable interactions and process constraints. The most common control method that can handle these challenges is model predictive control (MPC). While MPC is increasingly gaining acceptance, several opportunities for advancements in MPC applications exist. For the sake of brevity, instead of providing a general overview of MPC formulations, we focus in this section on specific unaddressed issues with MPC designs.

In several industrial applications of MPC, a linear model is used, in part due to the simplicity of developing linear models and in part due to the computational ease with using linear models. In order to handle the resultant plant-model mismatch, robust MPCs and offset-free MPC approaches have been developed. In robust MPC approaches, the control action is computed to handle the worst case effect of the uncertainty [49, 63]. These include Lyapunov-based MPC which enables explicit characterization of the region from where stability of the closed loop system under MPC controller

is achievable in the presence of constraints and uncertainty [46]. In another approach, the so-called offset-free MPC, the nominal model is integrated with augmented disturbance states to eliminate offset in set-point tracking [57, 70].

While these approaches are often able to eliminate uncertainty at steady state operation, the closed-loop performance certainly stands to improve if a better model is utilized in the control design. To determine if the closed-loop system is behaving as expected, existing approaches have focused on the area of control performance monitoring. In this direction, numerous MPC performance assessment methods are proposed to monitor the closed-loop performance by comparing the controller with a benchmark [11, 28, 65]. Most of these methods focus on tuning of the controller parameters to remedy the performance degradation. In model predictive approaches, where the control action is more directly dependent on the underlying model, there exists the necessity of explicitly monitoring model validity.

There exist some results on MPC with re-identification (IMPC) where model validity is accounted for by requiring excitation constraints to ensure that the model parameters remain identifiable [22]. In this approach, identification is performed at every time step. Furthermore, the approach requires finding the right trade-off between the inevitable performance deterioration (due to excitation conditions) and the possibility of loss of model validity. In [59] MPC Relevant Identification (MRI) was extended to Enhanced Multi-step Prediction Error Method (EMPEM). In [26] single input single output IMPC was extended to improve performance of the output regulation by not disturbing the plant when the model is deemed to have an acceptable precision. Acceptable precision is quantified through bounds on the variance of parameter estimates and parameter convergence rate in the MPC cost function. Explicit model monitoring based MPC designs for batch and continuous operation, however, are not currently available.

In another direction, it is common to use the same process for different products in chemical plants, with the different products achieved, say via grade transition in polymerization reactors.

This product transition is usually done by set-point change for plant output[54, 66, 72]. In these instances, a 'desired' process behavior could be specified as one where the transition to the new specification is the fastest, and a resultant optimization problem that minimizes the transition time is formulated and implemented [31].

An MPC framework has recently been proposed that enables specifying desired closed-loop behavior in more general terms for linear MIMO systems subject to input constraints [70], which is then implemented in conjunction with offset-free model predictive control. The developed approach [70] considers systems that are invertible (i.e. the inputs can be explicitly computed). A similar approach was utilized in [73] for linear systems. There does not exist a formulation, however, that allows the ability to explicitly prescribe the nature of the closed-loop behavior and have the formulation determine the best achievable closed-loop behavior for nonlinear systems.

With regard to stability characterization of MPC, in early MPC designs, the objective function was often utilized as a parameter to ensure closed-loop stability. In subsequent contributions, Lyapunov-based MPC was proposed where feasibility and stability from a well characterized region was built into the MPC [46, 50]. With increasing recognition (and ability) of MPC designs to focus on economic objectives, the notion of Economic MPC (EMPC) was developed for linear and nonlinear systems [4, 7, 40], and several important issues (such as input rate-of-change constraint and uncertainty) addressed. The key idea with the EMPC designs is the fact that the controller is directly given the economic objective to work with, and the controller internally determines the process operation (including, if needed, a set point) [53]. Most of the existing MPC formulations, economic or otherwise, have been illustrated using first principles models. In a recent result, an EMPC using empirical model was proposed [1]. The approach relies on a linearization approach, resulting in closed-loop stability guarantees for regions where the plant-model mismatch is sufficiently small, and illustrate results on stabilization around nominally stable equilibrium points. In summary, data driven MPC or EMPC approaches, that utilize appropriate modeling techniques to identify data

from closed-loop tests to handle operation around nominally unstable equilibrium points remain addressed.

1.2 Research Objectives and Thesis Outline

In Chapter 2, first, we address the problem of plant model mismatch by developing a model monitoring and closed-loop re-identification based MPC design. In the chapter, first, the general mathematical description for the systems considered in this work, and a representative formulation for linear model predictive control are presented. Then the proposed approach for closed-loop re-identification of plant is explained. The efficacy of the proposed method is illustrated through formulations and implementations for a nonlinear polymerization continuous stirred-tank reactor (CSTR) with input rate of change constraints and measurement noise.

In Chapter 3, we address the problem of plant model mismatch monitoring and evaluation and re-identification based MPC design for batch processes. First, the general description for the batch systems considered in this work, and a representative formulation for linear model predictive control are illustrated. Then the re-identification approach of plant is explained. The efficacy of the proposed method is illustrated through formulations and implementations for a nonlinear polymerization continuous stirred-tank reactor (CSTR) with input rate of change constraints and measurement noise.

In Chapter 4, we consider the problem of data driven model based economic MPC design. To this end, first a data-driven Lyapunov-based MPC is designed, and shown to be capable of stabilizing a system at an unstable equilibrium point. The data driven Lyapunov-based MPC utilizes an LTI model cognizant of the fact that the training data, owing to the unstable nature of the equilibrium point, has to be obtained from closed-loop operation or experiments. Simulation results are first presented demonstrating closed-loop stability under the proposed data-driven Lyapunov-based MPC. The

underlying data-driven model is then utilized as the basis to design an economic MPC. The economic improvements yielded by the proposed method are illustrated through simulations on a nonlinear chemical process system example.

In Chapter 5, we address the problem of control design for nonlinear systems that allows prescribing and determining the best achievable closed-loop behavior of a desired nature. First, the general mathematical description for the types of nonlinear systems considered in this work, and a representative formulation for nonlinear model predictive control (NMPC) are presented. Then the proposed bi-layer performance specification based nominal MPC scheme for achieving desired trajectories is given. The proposed framework enables specifying a desired nature of the closed-loop behavior and then determining the optimal feasible implementation of such behavior. Rigorous feasibility and stability properties are established for a formulation to achieve the best first order trajectory. Other formulations are also presented that demonstrate how, say a second order trajectory and input rate of constraints can be accommodated. The efficacy of the proposed method is first illustrated through formulations and implementations for a linear system subject to output feedback and a nonlinear continuous stirred-tank reactor (CSTR) with input rate of change constraints and uncertainty and a reactor separator plant.

Finally, Chapter 6 summarizes the main contributions of the research work, and recommendations for related future work and research opportunities are presented.

Chapter 2

MODEL PREDICTIVE CONTROL WITH CLOSED-LOOP RE-IDENTIFICATION[†]

Abstract

In this work we address the problem of handling plant-model mismatch by designing a subspace identification based MPC framework that includes model monitoring and closed-loop identification components. In contrast to performance monitoring based approaches, the validity of the underlying model is monitored by proposing two indexes that compare model predictions with measured past output. In the event that the model monitoring threshold is breached, a new model is identified using an adapted closed-loop subspace identification method. To retain the knowledge of the nominal system dynamics, the proposed approach uses the past training data and current input, output and

[†]The results in this chapter have been published in[36]:

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

set-point as the training data for re-identification. A model validity mechanism then checks if the new model predictions are better than the existing model, and if they are then the new model is utilized within the MPC. The effectiveness of the proposed method is illustrated through simulations on a nonlinear polymerization reactor.

Keywords: system identification, subspace identification, closed-loop identification, model predictive control, re-identification

2.1 Introduction

The operation of chemical plants faces numerous challenges such as inherent nonlinearity, complex variable interactions and process constraints. The most common control method that can handle these challenges is model predictive control (MPC). In several industrial applications of MPC, a linear model is used, in part due to the simplicity of developing linear models and in part due to the computational ease with using linear models. In order to handle the resultant plant-model mismatch, robust MPCs and offset-free MPC approaches have been developed.

In robust MPC approaches, the control action is computed to handle the worst case effect of the uncertainty [49, 63]. These include Lyapunov-based MPC which enables explicit characterization of the region from where stability of the closed loop system under MPC controller is achievable in the presence of constraints and uncertainty [46]. In another approach, the so-called offset-free MPC, the nominal model is integrated with augmented disturbance states to eliminate offset in set-point tracking [57, 70].

While these approaches are often able to eliminate uncertainty at steady state operation, the closed-loop performance certainly stands to improve if a better model is utilized in the control design. To determine if the closed-loop system is behaving as expected, existing approaches have focused on the area of control performance monitoring. In this direction, numerous MPC performance assessment

methods are proposed to monitor the closed-loop performance by comparing the controller with a benchmark [11, 28, 65]. Most of these methods focus on tuning of the controller parameters to remedy the performance degradation. In model predictive approaches, where the control action is more directly dependent on the underlying model, there exists the necessity of explicitly monitoring model validity.

There exist some results on MPC with re-identification (IMPC) where model validity is accounted for by requiring excitation constraints to ensure that the model parameters remain identifiable [22]. In this approach, identification is performed at every time step. Furthermore, the approach requires finding the right trade-off between the inevitable performance deterioration (due to excitation conditions) and the possibility of loss of model validity. In [59] MPC Relevant Identification (MRI) was extended to Enhanced Multi-step Prediction Error Method (EMPEM). In [26] single input single output IMPC was extended to improve performance of the output regulation by not disturbing the plant when the model is deemed to have an acceptable precision. Acceptable precision is quantified through bounds on the variance of parameter estimates and parameter convergence rate in the MPC cost function.

In particular, in this method, autoregressive models with exogenous inputs are used and recursive weighted least-squares algorithm is utilized to estimate model parameters. In order to solve the trade-off between control performance and persistence of excitation, in a recent contribution [10, 58] maximizing the MPC objective function is used instead of minimization to maximize signal variance and address the feasibility and stability of MPC with re-identification.In another direction, in [5], plant-model mismatch is detected by partial correlation analysis in order to determine the correlation between model residual of each output and each manipulated variable with effect of disturbance and other manipulated variables removed. This correlation may be significant in the presence of plant-model mismatch. In [6] model-plant mismatch is quantified by comparison of actual and achieved control quality. Note that in theses studies it is assumed that sufficient set-point excitation

is available in order to calculate the plant-model mismatch. In the proposed approach, the set point excitation is not necessary for the monitoring aspect. Furthermore, in these approaches, the original training data is not retained in the new model identification, and these methods are designed to address situations where the system is changed significantly and previous data are not at all representative of the plant in question. In situations where plant model mismatch arises due to change in operating condition (with the possibility of reverting back to the nominal plant operation), it becomes useful to merge old and new plant data in the re-identification step.

Motivated by the above considerations, in this work we address the problem of plant model mismatch by developing a model monitoring and closed-loop re-identification based MPC design. The rest of the manuscript is organized as follows: First, the general mathematical description for the systems considered in this work, and a representative formulation for linear model predictive control are presented. Then the proposed approach for closed-loop re-identification of plant is explained. The efficacy of the proposed method is illustrated through formulations and implementations for a non-linear polymerization continuous stirred-tank reactor (CSTR) with input rate of change constraints and measurement noise. Finally, concluding remarks are presented.

2.2 Preliminaries

In this section, a brief description of the general class of processes that are considered in this study is provided. Then, the orthogonal projection based subspace identification and a representative MPC formulation is presented.

2.2.1 Problem Statement

Consider a general multi-input multi-output (MIMO) controllable system, with $y \in \mathbb{R}^{n_y}$ denoting the measured outputs, and $u \in \mathbb{R}^{n_u}$ denoting the vector of constrained control (manipulated) input variables, 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. In keeping with the discrete implementation of MPC, 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.

Subspace Identification

In this section the conventional state space subspace identification method is reviewed [27, 67, 76]. In the subspace identification approach, the goal is to determine the system matrices for a discrete linear time invariant model of the following form:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{2.1}$$

$$y_k = Cx_k + Du_k + v_k \tag{2.2}$$

where $x \in \mathbb{R}^{n_x}$ denotes the vector of state variables, $y \in \mathbb{R}^{n_y}$ denotes the vector of measured outputs, $w \in \mathbb{R}^{n_x}$ and $v \in \mathbb{R}^{n_y}$ are zero mean, white vectors of process noise and measurement noise with the following covariance matrices:

$$E[\begin{pmatrix} w_i \\ v_j \end{pmatrix} \begin{pmatrix} w_i^T & v_j^T \end{pmatrix}] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{ij} \tag{2.3}$$

where $Q \in \mathbb{R}^{n_x \times n_x}$, $S \in \mathbb{R}^{n_x \times n_y}$ and $R \in \mathbb{R}^{n_y \times n_y}$ are covariance matrices, and, δ_{ij} is the Kronecker delta function. To identify the system matrices, Hankel matrices are first constructed by stacking the process variables as follows:

$$U_p = U_{1|i} = \begin{bmatrix} u_1 & u_2 & \dots & u_j \\ u_2 & u_3 & \dots & u_{j+1} \\ \dots & \dots & \dots & \dots \\ u_i & u_{i+1} & \dots & u_{i+j-1} \end{bmatrix} \tag{2.4}$$

$$U_f = U_{i+1|2i} = \begin{bmatrix} u_{i+1} & u_{i+2} & \dots & u_{i+j} \\ u_{i+2} & u_{i+3} & \dots & u_{i+j+1} \\ \dots & \dots & \dots & \dots \\ u_{2i} & u_{2i+1} & \dots & u_{2i+j-1} \end{bmatrix} \tag{2.5}$$

where U_p and U_f denote the past and future input Hankel matrices. i is a user-specified parameter that limits the order of the system (n) (which in itself is a user-specified parameter). Similar block-Hankel matrices are made for output, process and measurement noises $Y_p, Y_f, V_p, V_f \in \mathbb{R}^{in_y \times j}$ and $W_p, W_f \in \mathbb{R}^{in_x \times j}$ are defined in the similar way. The state sequences are defined as follows:

$$X_p = \begin{bmatrix} x_1 & x_2 & \dots & x_j \end{bmatrix} \tag{2.6}$$

$$X_f = \begin{bmatrix} x_{i+1} & x_{i+2} & \dots & x_{i+j} \end{bmatrix} \tag{2.7}$$

furthermore with:

$$\Psi_p = \begin{bmatrix} Y_p \\ U_p \end{bmatrix} \tag{2.8}$$

$$\Psi_f = \begin{bmatrix} Y_f \\ U_f \end{bmatrix} \tag{2.9}$$

The orthogonal projection of row space of matrix A onto row space of matrix B, (A/B) is defined as:

$$A/B = AB^{\dagger}B \tag{2.10}$$

where the superscript † stands for pseudo-inverse. By recursive substitution into the state space model equations Eqs. (2.1,2.2), it is straightforward to show:

$$Y_f = \Gamma_i X_f + \Phi_i^d U_f + \Phi_i^s W_f + V_f \tag{2.11}$$

$$Y_p = \Gamma_i X_p + \Phi_i^d U_p + \Phi_i^s W_p + V_p \tag{2.12}$$

$$X_f = A^i X_p + \Delta_i^d U_p + \Delta_i^s W_p \tag{2.13}$$

where:

$$\Gamma_i = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{bmatrix} \tag{2.14}$$

$$\Phi_i^d = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & \dots & D \end{bmatrix} \tag{2.15}$$

$$\Phi_i^s = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ C & 0 & 0 & \dots & 0 & 0 \\ CA & C & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & 0 & 0 \\ CA^{i-2} & CA^{i-3} & CA^{i-4} & \dots & C & 0 \end{bmatrix} \tag{2.16}$$

$$\Delta_i^d = \begin{bmatrix} A^{i-1}B & A^{i-2}B & \dots & AB & B \end{bmatrix} \tag{2.17}$$

$$\Delta_i^s = \begin{bmatrix} A^{i-1} & A^{i-2} & \dots & A & I \end{bmatrix} \tag{2.18}$$

Eq. (2.11) can be rewritten in the following form to have the input and output data at the LHS of the equation([71]):

$$\begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \begin{bmatrix} Y_f \\ U_f \end{bmatrix} = \Gamma_i X_f + \Phi_i^s W_f + V_f \tag{2.19}$$

By orthogonal projecting of Eq. (2.19) onto Ψ_p:

$$\begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \Psi_f / \Psi_p = \Gamma_i X_f / \Psi_p + \Phi_i^s W_f / \Psi_p + V_f / \Psi_p \tag{2.20}$$

The last two terms in RHS of Eq. 2.20 are orthogonal projection of the future noise onto the row space of Ψ_p, and since the noise terms are independent, these two term are equal to zero. Thus Eq. (2.20) is simplified as follows:

$$\begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \Psi_f / \Psi_p = \Gamma_i X_f / \Psi_p \tag{2.21}$$

Equation (2.21) indicates that the column space of Γ is equal to column space of LHS of the Equation (2.21), and the row space of X_f/Ψ_p is the same as the row space of left hand side of the equation [27, 29].This equation can be solved using singular value decomposition (SVD), and the system matrices can be calculated from the results. This is the essence of the open-loop subspace identification method which assumes that future inputs are independent of the future disturbances. Note that this assumption does not hold for data under closed-loop control, and can result in biased results [29]. To deal with this, the subspace identification approach was adapted by utilizing a new variable, denoted as an instrument variable, as part of the identification procedure. The new instrument variable, that satisfies the independence requirement, is used to project both side of the Equation 2.20 and the result is used to determine LTI model matrices. In existing results, the innovation form of the LTI model is used, as follows:

$$x_{k+1} = Ax_k + Bu_k + Ke_k \tag{2.22}$$

$$y_k = Cx_k + Du_k + e_k \tag{2.23}$$

where e_k is the innovation term, and, K is filter gain. In these methods, after determining the A matrix, B and K are estimated using least squares. In contrast to the existing results, in this

work we first estimate the noise terms, then calculate noise covariance matrices and based on these covariances the observer gain is calculated.

The system identification procedures require input signal to be quasi-stationary and persistently exciting of order $2i$ [41, 60]. In the system identification step, only the observable part of a system is identified therefore the LTI model is always observable. Also note that, the order of the LTI system n is selected in a way that the identified system is controllable and the prediction of validation data is acceptable.

2.2.2 Model Predictive Control

The MPC formulation uses the identified model to determine the optimum control action. In order to handle plant-model mismatch, often an offset-free mechanism is utilized. Thus the identified model is augmented with integrating disturbances, d as follows.

$$\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k \tag{2.24}$$

$$y_k = \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + Du_k \tag{2.25}$$

where $d \in \mathbb{R}^{n_d}$ is the vector of augmented disturbances, and B_d and C_d are matrices with appropriate dimensions. The MPC computations require estimation of the subspace and disturbance states. One way to estimated these is to use a Luenberger observer as follows:

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k - L(y_k - \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} - Du_k) \tag{2.26}$$

where L is predictor gain matrix. In this study, a Luenberger observer is utilized to estimate LTI model states and augmented disturbance states. Also note that other kinds of state estimators such as Kalman filter or moving horizon estimation (MHE) can be used in the proposed framework.

In the Offset-free MPC at each sample time l, the control action is computed as follows:

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

subject to:

$$\begin{aligned} &\tilde{x}_{k+1} = A\tilde{x}_k + B_d\tilde{d}_k + B\tilde{u}_k \\ &\tilde{d}_{k+1} = \tilde{d}_k \\ &\tilde{y}_k = C\tilde{x}_k + C\tilde{d}_k + D\tilde{u}_k \\ &\tilde{u} \in \mathcal{U}, \qquad \Delta\tilde{u} \in \mathcal{U}_\delta, \qquad \tilde{x}(k) = \hat{x}_l \end{aligned} \tag{2.27}$$

where P denotes the prediction horizon, y^{SP}_k is the desired output (asymptotically constant reference signal), and, $\tilde{y}_k$ is the predicted output trajectory at the time $k\Delta t$. $Q_y \in R^{n\times n}$ is a positive definite matrix, and, R_{du} is a positive semi-definite matrix, and they are chosen so the nominal closed-loop system is stable. For the conditions required to ensure offset free tracking, see, e.g, [43].

Remark 2.1. *Note that there are different versions of the offset-free MPC approach, including those where stability constraints are utilized. We present here a generic offset-free MPC approach to simply illustrate our model monitoring and re-identification approach.*

2.3 Model Monitoring based MPC Implementation

In this section, we present the model monitoring based MPC implementation. To this end, we first describe the model monitoring approach, followed by the trigger for re-identification. Subsequently,

a variation of the closed-loop subspace identification method is presented.

2.3.1 Model Monitoring and Re-Identification Trigger

An MPC computes the control action by minimizing an objective function along a prediction horizon, by predicting the output trajectory using the current state of the system, and candidate input moves. The validity of the process model is therefore critical to, although not the only factor in achieving good closed-loop performance. Most of the existing approaches focus on control performance monitoring. In the present manuscript, we instead directly focus on the model prediction error as one of the root causes of performance degradation. In this section, we describe an approach that monitors the health of the process model, and propose a couple of triggers for re-identification.

The key idea behind model monitoring is to compare model predictions with observed behavior. Before we present the details of the proposed method, let us consider other possibilities that will not allow a model validation online. The first of these recognitions is that at any point in time, predicting ahead (either as part of the MPC or otherwise), will not provide any information about model validity, because future process variables are simply not yet available. Therefore, the only recourse to assess the model prediction is to 'go back' to a past data point, and predict ahead up to the current time. The ability to 'go back to a past data point' however, depends on several factors.The first depends on the nature of the model. If the model being utilized in the MPC is a nominal model (without augmented states), and state measurements are available, then starting from the measured states in the past, and utilizing the known input values, the predictions can be computed, and compared with process measurements. Furthermore, even in that case, the trajectory computed by the MPC at the past point in time can not be utilized as the basis for comparison, because the actual input trajectory implemented on the plant will likely be different due to the feedback and receding horizon implementation of MPC. For cases where a subspace model is being utilized (along with augmented disturbance states), then further care should be taken to enable an

appropriate model validity comparison, primarily by ensuring that the best known values of the state estimates (known at the point in time in the past) are utilized instead of recomputing those (as with moving horizon estimation approaches). See Remark 2 for further discussion on this.

A schematic is presented in Fig. 2.1 to illustrate the monitoring approach. In particular, model monitoring at a sample time k is achieved by going back to the sample time $k - P$, where P is the chosen 'lookback' horizon and using the state estimate generated by the state estimator at time step $k - P$ ($\hat{x}_{k-P}$) and inputs from $k - P$ to $k - 1$ to predict the output trajectory from $\bar{y}_{k-P}$ to $\bar{y}_{k-1}$. Then the predicted past output trajectory is compared with the measured output of plant and based on this comparison, the model accuracy is evaluated.

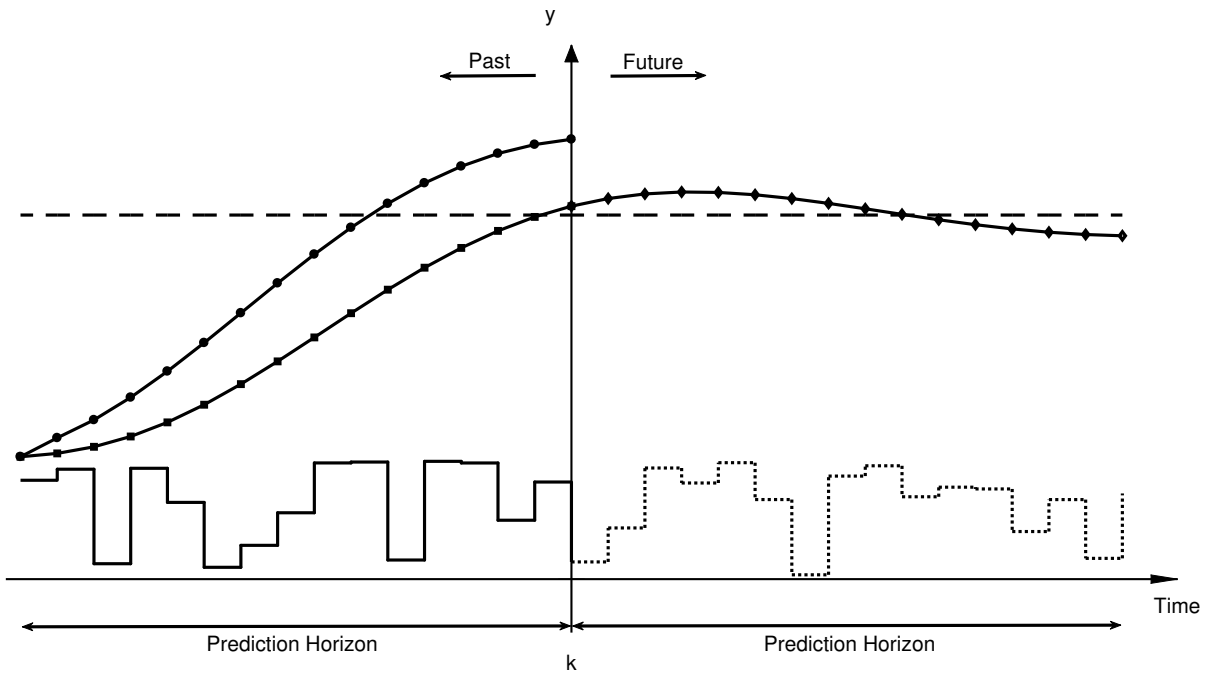

Figure 2.1: MPC with model prediction monitoring scheme([Measured Output:−■−], [Past Predicted Output:−●−], [Future Predicted Output:−◆−], [Setpoint:−−], [Past Input:−], [Future Predicted Input:..])

In the schematic in Figure 2.1, the line (−●−) is the predicted output trajectory that is calculated using the augmented identified LTI model with estimated state at the sample time $k - P$ (stored at

each sample time for the purpose of model monitoring) and the implemented input from $k - P$ to $k - 1$ (the continuous line(−)). The measured outputs of the system are shown by the line [−■−]. In the deterministic case with no model mismatch, the measured output and the predicted output by the MPC model would be exactly the same. Due to model mismatch, these two lines would be different. The dashed line(−) is the set-point. The dotted line (..) is the future predicted input and the line (−◆−) is the future predicted output, calculated by the MPC at the current time step. Note, that the MPC calculations at the current time step are not required for the model monitoring at the current time step. Finally, the model monitoring computations do not require the solution of any optimization problem.A block diagram of the MPC with closed-loop re-identification approach is shown in Fig. 2.2. At the re-identification step, models with the number of states close to the original model order should be explored, and among the observable and controllable models, the model which has the best prediction on the validation data set should be chosen as the new model. In order to evaluate model accuracy at each sample time l, the updated state x_{l-P} which was stored after calculation by state estimator at sample time $l - P$ and implemented input trajectory from u_{l-P} to u_{l-1} are utilized as follows:

$$\begin{aligned} \bar{y}_j &= C\bar{x}_j + Du_j \\ \bar{x}_{j+1} &= A\bar{x}_j + Bu_j \\ \text{where } j &= l - P, \ldots, l - 1 \\ \bar{x}_l &= \hat{x}_{l-P} \end{aligned} \tag{2.28}$$

where $\bar{y}$ and $\bar{x}$ are predicted output and state. Having obtained the predictions for the process outputs, more than one indexes are possible to be formulated to represent model accuracy. Here,

two illustrative indexes are proposed to evaluate MPC model prediction:

$$MPE_k = \frac{1}{P} \sum_{j=k-P}^{k-1} (|y_j - \bar{y}_j|)$$
$$EPPE_k = |y_{k-1} - \bar{y}_{k-1}| \tag{2.29}$$

where $MPE_k \in R^{n_y}$ is the vector of mean prediction error, which is the mean value of prediction error along the past horizon at sample time k, $EPPE_k \in R^{n_y}$ is the vector of end-point prediction error at sample time k, and, $\bar{y}$ is the corresponding output prediction.

The re-identification triggers when the MPC model performance index exceeds a specific threshold. This threshold can be set using training/validation data of the initial model by using Eq. (2.29) with the data. The index is calculated at each sample time and compared with the threshold to trigger re-identification. For re-identification, the recent data needs to be appropriately augmented with the past data to obtain the model. The details are presented in the next section.

After re-identification, a new observer gain is calculated. Then the new observer will utilize recent data to estimate current state to have the most up to date state to predict the plant behavior in MPC.

Remark 2.2. *Note that both the proposed approach and the moving horizon estimation (MHE), which is a state estimation technique, utilize past output and input data, albeit with different objectives and in different ways. In MHE the past input, output and model are used for the purpose of state estimation, where the state is estimated to minimize (along with other terms) the error between the observed and the predicted output. In contrast, the proposed approach does not change, but uses the state estimate that was computed at the past time instance, to predict forward up to the current time, and utilizes the difference between predicted and observed outputs as a model monitoring index.*

2.3.2 Closed-Loop Identification

Having formulated the model monitoring and triggering approach, the next step is the utilization of an appropriate identification method that can handle the nature of data availability. As mentioned earlier, most of the open-loop identification methods are based on the assumption that input and process disturbances are not correlated [18, 61, 68]. Under feedback control, the future input is correlated with noise signals, which may result in biased estimation of the model in the standard open-loop identification methods [60].

The following short-hand notation is used in closed-loop method:

$$\Psi_{pr} = \begin{bmatrix} R_f \\ \Psi_p \end{bmatrix} \tag{2.30}$$

where R_f is data-Hankel matrix of set-point. Huang et al. in [29] showed that by using Ψ_{pr} as an instrument subspace variable, the bias error of open-loop methods for closed-loop data can be avoided. Therefore by projecting Eq. (2.19) onto Ψ_{pr}:

$$\begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \Psi_f / \Psi_{pr} = \Gamma_i X_f / \Psi_{pr} + \Phi_i^s W_f / \Psi_{pr} + V_f / \Psi_{pr} \tag{2.31}$$

Since the future process and measurement noises are independent of the past input/output and future setpoint Eq. (2.20), the noise terms would be equal to zero, and the resultant equation would have the following form:

$$\begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \Psi_f / \Psi_{pr} = \Gamma_i X_f / \Psi_{pr} \tag{2.32}$$

By multiplying Eq. (2.32) by the extended orthogonal observability $\Gamma_i^{\perp}$, the state term is eliminated:

$$(\Gamma_i^{\perp})^T \begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \Psi_f/\Psi_{pr} = 0 \tag{2.33}$$

Therefore the column space of Ψ_f/Ψ_{pr} is orthogonal to the row space of $\begin{bmatrix} (\Gamma_i^{\perp})^T & -(\Gamma_i^{\perp})^T\Phi_i^d \end{bmatrix}$. By performing singular value decomposition (SVD) of Ψ_f/Ψ_{pr}:

$$\Psi_f/\Psi_{pr} = U\Sigma V = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {V_1}^T \\ {V_2}^T \end{bmatrix} \tag{2.34}$$

where Σ_1 contains dominant singular values of Ψ_f/Ψ_{pr} and theoretically it has the order of the $n_u i + n$ and the order of the system can be determined by the number of the dominant singular values of the Ψ_f/Ψ_{pr} [71]. The orthogonal column space of Ψ_f/Ψ_{pr} is $U_2 M$, where $M \in \mathbb{R}^{(n_y-n)i\times(n_y-n)i}$ is any constant nonsingular matrix and is typically chosen as an identity matrix. Huang et al. in [29], proposed the following steps to solve for the LTI model:

$$(\begin{bmatrix} \Gamma_i^{\perp} & -\Gamma_i^{\perp}\Phi_i^d \end{bmatrix})^T = U_2 M \tag{2.35}$$

From Eq.(2.35), Γ_i and Φ_i^d can be estimated.

$$\begin{bmatrix} {\Gamma_i}^{\perp} \\ -(\Phi_i^d)^T{\Gamma_i}^{\perp} \end{bmatrix} = U_2 \tag{2.36}$$

which results in (using MATLAB matrix index notation):

$$\begin{cases} \hat{\Gamma}_i = U_2(1 : n_y i, :)^{\perp} \\ \hat{\Phi}_i^d = -(U_2(1 : n_y i, :)^T)^{\dagger} U_2(n_y i + 1 : end, :)^T \end{cases} \tag{2.37}$$

The past state sequence can be calculated as follows:

$$\hat{X}_i = \hat{\Gamma}_i^{\dagger} \begin{bmatrix} I & -\hat{\Phi}_i^d \end{bmatrix} \Psi_f / \Psi_{pr} \tag{2.38}$$

The future state sequence can be calculated by changing data Hankel matrices as follows [29]:

$$R_f = R_{i+2|2i} \tag{2.39}$$

$$U_p = U_{1|i+1} \tag{2.40}$$

$$Y_p = Y_{1|i+1} \tag{2.41}$$

$$U_f = U_{i+2|2i} \tag{2.42}$$

$$Y_f = Y_{i+2|2i} \tag{2.43}$$

$$\Rightarrow \hat{X}_{i+1} = \hat{\underline{\Gamma}}_i^{\dagger} \begin{bmatrix} I & -\underline{\hat{H}}_i^d \end{bmatrix} \Psi_f / \Psi_{pr} \tag{2.44}$$

where $\hat{\underline{\Gamma}}_i$ is obtained by eliminating the last n_y rows of Γ_i, and $\underline{H}_i^d$ is obtained by eliminating the last n_y rows and the last n_u columns of H_i^d. Then the model matrices can be estimated using least square:

$$\begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} W_{i|i} \\ V_{i|i} \end{bmatrix} \tag{2.45}$$

The system matrices can be calculated as follows:

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix}^{\dagger} \tag{2.46}$$

As mentioned before, the utilized closed-loop method is based on the recently presented subspace orthogonal projection identification method (CSOPIM) [29]. The difference between these two

methods is that, in CSOPIM the innovation form of the LTI model is used to formulate the identification method. In CSOPIM, first the innovation term E is estimated using residual of measured output and estimated output ($CX_i + DU_{i|i}$), then the innovation gain (K) is calculated using least square. This method may cause a situation where $A - KC$ of the identified model ends up having eigenvalues out of the unit circle, therefore the LTI model with filter is not going to be stable. In order to avoid this problem, the presented approach uses a discrete-time linear time invariant state-space model for the closed-loop formulation and covariance matrices are calculated using the estimated residuals of state and output prediction instead of incorporating K in the LTI model.

With the proposed approach, process and measurement noise Hankel matrices can be calculated as the residual of the least square of Eq. 2.45:

$$\begin{bmatrix} \hat{W}_{i|i} \\ \hat{V}_{i|i} \end{bmatrix} = \begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix} \tag{2.47}$$

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

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = E(\begin{bmatrix} \hat{W}_{i|i} \\ \hat{V}_{i|i} \end{bmatrix} \begin{bmatrix} \hat{W}_{i|i}^T & \hat{V}_{i|i}^T \end{bmatrix}) \tag{2.48}$$

Note, that any closed-loop identification method which can handle batches of data with unequal durations (for instance, closed-loop predictor-based subspace identification, see, e.g., [14]) can be used in the proposed framework. The specific identification method used in the present paper is chosen because it is non-iterative and no knowledge of the controller dynamics is required.

2.3.2.1 Formation of Batch Data Hankel Matrices

The proposed re-identification method intends to utilize past training data (used to create initial model) and recent plant data. In order to assimilate recent data with past training data, the different data sets are recognized as different batches of process data. Since these data are not continuous block of data, these data can not be handled as one block of data. The way these batches of data are handled is that the Hankel matrices for each single batch is concatenated horizontally and thus the pertinent properties for subspace identification are retained [62]. For each batch of data b with input variable define the Hankel sub-matrix as:

$$U_p^{(b)} = U_{1|i}^{(b)} = \begin{bmatrix} u_1^{(b)} & u_2^{(b)} & \dots & u_{j^{(b)}}^{(b)} \\ \vdots & \vdots & \vdots & \vdots \\ u_i^{(b)} & u_{i+1}^{(b)} & \dots & u_{i+j^{(b)}-1}^{(b)} \end{bmatrix} \tag{2.49}$$

The overall pseudo-Hankel matrix is formed by concatenating Hankel matrices horizontally:

$$U_p = \begin{bmatrix} U_p^{(1)} & U_p^{(2)} & \dots & U_p^{(B)} \end{bmatrix} \tag{2.50}$$

where B is the total number of the data batches. The described system identification utilizes the concatenated data-Hankel matrices created as explained in Eq. 2.50 to identify an LTI model for MPC.

Remark 2.3. *Note that data-Hankel matrices can not be used in the form of Equation 2.49 because it can only include one batch of data. Also, the other batch of data can not be added to the initial Hankel matrix because the identification method would treat the entire data as one single block of data, which would be incorrect. Therefore in order to create data-Hankel matrices without requiring that the end point of one batch of data is the beginning of next batch of data, Hankel matrices are created with horizontally concatenating data-Hankel matrices from the original data set and the*

recent data (y, u and r).

Remark 2.4. *Note, that in contrast to existing re-identification methods, this framework enables using prior training data, together with new data, in the identification of the new model. This allows (and recognizes the fact), that the process may not have changed entirely, but is simply operating in a region where the control calculations will be better served by tweaking the original model (but not abandoning the original model altogether). One of the benefits of the proposed approach is that if the initial training data satisfies persistently excitation condition for a certain system order, then the new concatenated data will also be persistently exciting for that system order.*

Remark 2.5. *After re-identification the new identified model is augmented with disturbance states and state estimator is designed for the new augmented system. Since the MPC objective function and constraints only involves input and output terms, retuning of controller is not necessary.*

Remark 2.6. *In the case of a occasionally occurring disturbances, such as a 'gate-shaped' disturbance, if the disturbance lasts long enough to trigger re-identification, a new model will be identified but after the disturbance passes, the model prediction index would violate the threshold and a new model would be identified again, which would be more similar to the initial model. If the disturbance keeps happening, this procedure would keep repeating. Note that this would happen with any other adaptive model based control design as well. Repeat instances of such behavior would indicate that the 'time constant' of large process disturbances is of the same order as the 'time constant' for the identified model, in turn necessitating the identification of the model at a smaller time scale.*

Remark 2.7. *In order to achieve smooth, bump-less, transition after re-identification, two mechanisms are utilized in the present approach. The first is that recent data is utilized to first update the states of the new model, at the time of switching in the new model. This avoids the initial fluctuation of the state estimation by the observer, in turn avoiding a jump in input calculation by controller. The second mechanism is the use of constraints on the rate of change of the input.*

Remark 2.8. *The initial training data would be used in all of the re-identification steps, and since*

the identification method is capable of handling different batches of data, the windows of data which indicated poor prediction of the previous models can also be included for the re-identification purpose, thus ensuring that the new model better captures the new changed dynamics. In order to handle memory issues, the more recent data, if sufficiently rich, may be used to replace the training data.

2.4 Illustrative Simulation Results

In this section, we implement the proposed MPC with closed-loop re-identification on a polymerization reactor example [10]. The state space equation of the reactor is nonlinear with 7 states, 2 outputs and 2 inputs. The list of the reactions that occurs in the reactor are presented in Table 2.1. The mathematical model of the dynamic system is as follows:

$$
\begin{aligned}
\dot{C}_I &= \frac{Q_i C_{I_f} - Q_t C_I}{V} - k_d C_I \\
\dot{C}_M &= \frac{Q_M C_{M_f} - Q_t C_M}{V} - k_p C_M C_P \\
\dot{T} &= \frac{Q_t (T_f - T)}{V} + \frac{(-\Delta H)}{\rho c_p} k_p C_M C_P - \frac{hA}{\rho c_p V}(T - T_c) \\
\dot{T}_c &= \frac{Q_t (T_{cf} - T_c)}{V_c} + \frac{hA}{\rho_c c_{cp} V_c}(T - T_c) \\
\dot{D}_0 &= 0.5 k_t C_p^2 - \frac{Q_t D_0}{V} \\
\dot{D}_1 &= M_m k_p C_M C_p - \frac{Q_t D_1}{V} \\
\dot{D}_2 &= 5 M_m k_p C_M C_p + \frac{3 M_M k_p^2}{k_t} C_M^2 - \frac{Q_t D_2}{V}
\end{aligned}
\tag{2.51}
$$

where

$$\begin{aligned} C_P &= (\frac{2 f_i k_d C_I}{k_t})^{0.5} \\ k_j &= A_j \exp{(\frac{-E_j}{T})}, j = d, p, t \\ Q_t &= Q_i + Q_s + Q_m \end{aligned} \tag{2.52}$$

The measured outputs are reactor temperature (T) and the intrinsic viscosity (η), related to the process states as:

$$\eta = 0.0012(M_m \frac{D_2}{D_1})^{0.71} \tag{2.53}$$

The manipulated variables are the cooling liquid flow rate (Q_c) and monomer feed flow rate (Q_m). The process parameters and steady state condition for the CSTR are presented in Table (2.2) and Table (2.3).

In order to identify the process model, PI controllers (pairing η with Q_c and T with Q_m) are implemented on the reactor. In particular, pseudo-random binary signals are used as set-point for PI controllers. The training data is shown in Figures (2.4 and 2.5).

The proposed variation on the closed-loop identification algorithm uses these data to identify an LTI model. The order of the identified LTI is selected as $n = 4$ and $i = 12$. Model validation results

Table 2.1: List of Reactions in the Reactor

No.	Reaction	Description
1	$I \xrightarrow{k_d} 2R$	Initiator Decomposition
2	$M + R \xrightarrow{k_i} P1$	Chain Initiation
3	$P_n + M \xrightarrow{k_p} 2R$	Propagation
4	$P_n + P_m \xrightarrow{k_{td}} T_n + T_m$	Termination by Disproportionation
5	$P_n + P_m \xrightarrow{k_{tc}} T_{n+m}$	Termination by Combination

Table 2.2: List of Process Parameters for the Polymerization Process [3]

Variable	Description	Units	Value
A_d	Frequency factor for initiator decomposition	h^{-1}	2.142×10^{17}
E_d	Activation energy for initiator decomposition	K	14897
A_p	Frequency factor for propagation reaction	L/(mol.h)	3.816×10^{10}
E_p	Activation energy for propagation reaction	K	3557
A_t	Frequency factor for propagation reaction	L/(mol.h)	3.816×10^{10}
E_t	Activation energy for termination reaction	K	843
f_i	Initiator efficiency	–	0.6
$-\Delta H_r$	Heat of polymerization	J/mol	6.99×10^4
hA	Overall heat transfer coefficient	J/(K.h)	1.05×10^6
ρc_p	Heat capacity of reactor fluid	J/(K.L)	1506
$\rho_c c_{pc}$	Heat capacity of cooling jacket fluid	J/(K.L)	4043
M_m	Molecular weight of the monomer	g/mol	104.14

under a new set of set-point changes is presented in Figure 2.3.

In the model validation step, initially a steady state Kalman filter is used to update state estimate until $t = 300hrs$ and after convergence of the states (gauged via convergence of the outputs), the model and the input trajectory (without the state estimator) is utilized to predict the future output. As shown in Figure 2.3 the model prediction is acceptable. This model is used as an initial model to design an MPC. In order to handle model mismatch, the model is augmented with disturbance states. The disturbance model in the simulation example is constructed with B_d chosen as B and C_d chosen as the identity matrix [69].

In order to make sure that the re-identification trigger is not overly sensitive to disturbance or measurement noise, or minor plant-model mismatch, the re-identification is triggered only when the error index threshold is violated for a certain number of conservative sample times. Note, that 4 consecutive sample times is specific to the present case study, and was chosen to avoid unnecessary re-identification (a smaller number resulted in too many re-identifications, while larger numbers made the approach sluggish, and did not yield performance improvements). For other applications, an appropriate number should similarly be chosen. The simulation results are presented

Table 2.3: List of Steady-State Operational Condition for the Polymerization Process [3]

Variable	Description	Units	Value
C_{I_f}	Concentration of initiator in feed	mol/L	0.5888
C_{M_f}	Concentration of monomer in feed	mol/L	8.6981
C_I	Concentration of initiator in the reactor	mol/L	$6.6832 \times 10^{-}2$
C_M	Concentration of monomer in the reactor	mol/L	3.3245
D_0	Molar concentration of dead polymer chains	mol/L	$6.7547 \times 10^{-}4$
D_1	Mass concentration of dead polymer chains	g/L	16.110
Q_i	Flow rate of initiator	L/h	108
Q_s	Flow rate of solvent	L/h	459
Q_m	Flow rate of monomer	L/h	378
Q_c	Flow rate of cooling jacket fluid	L/h	471.6
T_f	Temperature of reactor feed	K	330
T_{cf}	Inlet temperature of cooling jacket fluid	K	295
T	Temperature of the reactor	K	323.56
T_c	Temperature of cooling jacket fluid	K	305.17
V	Reactor volume	L	3000
V_c	Volume of cooling jacket fluid	L	3312.4

in Figures (2.6-2.9). At sample time $k = 57$ ($time = 28.5$) the model prediction error violated the specified ϵ_{MPE} threshold for model prediction accuracy (thresholds are reported in Table 2.4) and re-identification is triggered and the model is replaced with a new model with the same order. The new model also is augmented and the state of the new model is estimated using the recent data by utilizing the observer with recent data. The LTI model parameters are presented in Table 2.5.

For subsequent comparison of the proposed methods a cost variable defined as follows:

$$j = \sum_{k=t_1}^{Nt} ||y_k - y_k^{\mathrm{SP}}||_{Q_y}^2 + ||u_k - \tilde{u}_{k-1}||_{R_{du}}^2 \tag{2.54}$$

where t_1 is the sample time the re-identification is triggered, and N_t is the total number of sample times, the values of this variable is evaluated for five simulations with different measurement noise and are presented in Table 2.6. The realized objective function of MPC is shown in Figure. 2.10, and indicates an improved closed-loop performance. The summation of the stage cost over the

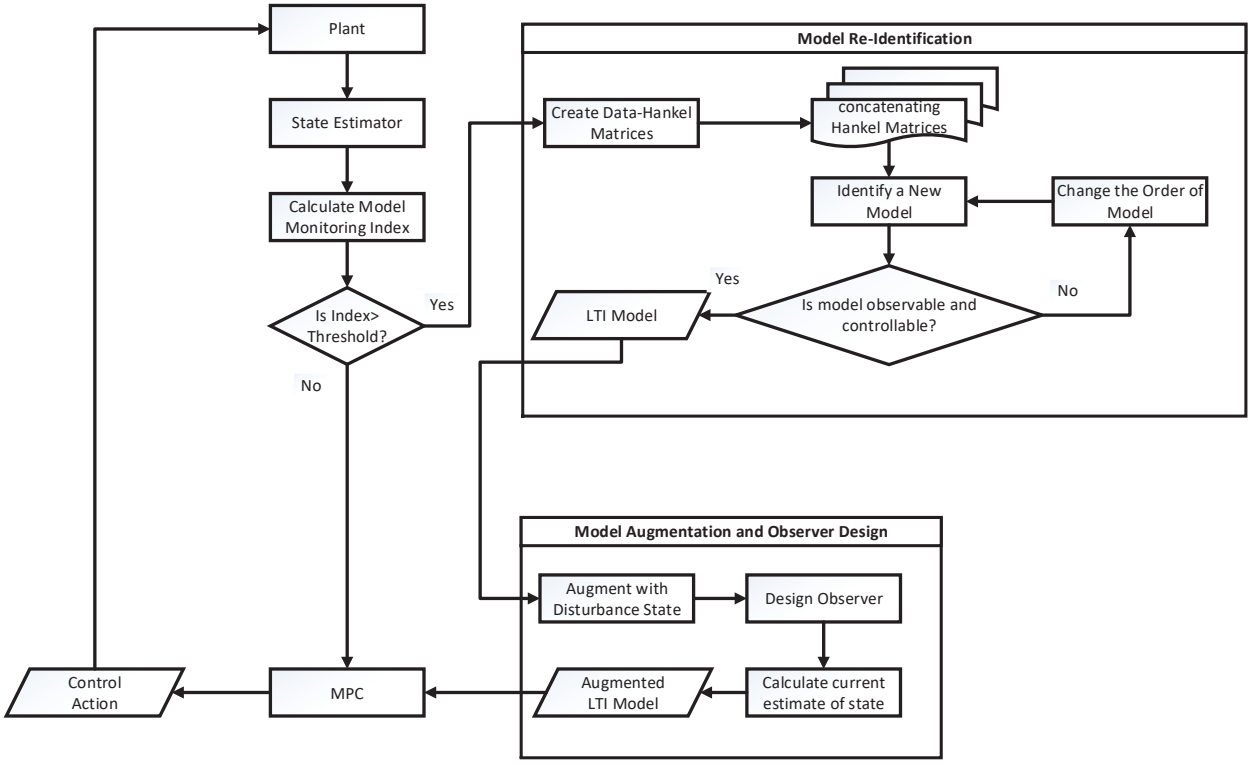

Figure 2.2: MPC with closed-loop re-identification

entire simulation time is used as a measure of closed-loop performance. The prediction error indexes in Figures (2.6 and 2.7) with re-identification decays faster than the case with no re-identification (except at some very few isolated points). Also, note that even without re-identification the error indexes decay because in the offset-free MPC, the LTI system is augmented with integrated disturbances to eliminate offset. With re-identification the prediction error, and in turn the closed-loop performance, improves faster because re-identifying 'jump starts' the offset-free mechanism with the updated model. As can be seen, the closed-loop behavior is improved wherein the oscillation of the outputs are reduced significantly, and input changes are less aggressive. Furthermore, the realized MPC objective function, which used the implemented input, and realized output of the plant to calculate the objective function is less than nominal MPC.

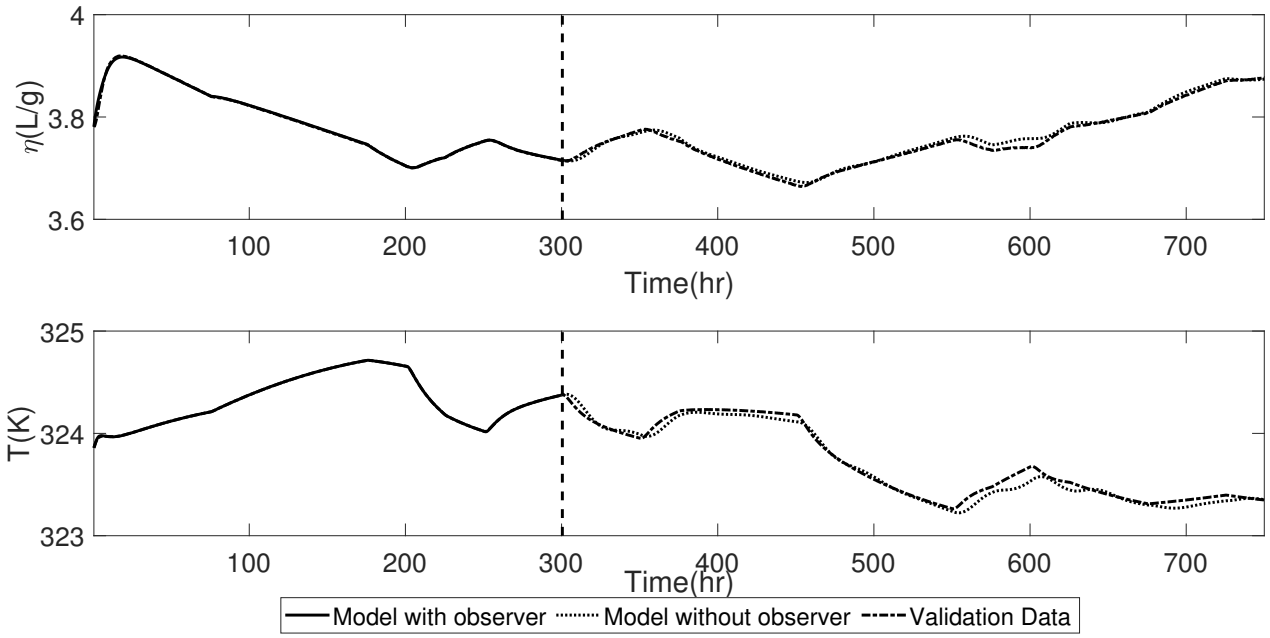

Figure 2.3: Model validation results under PI controller

2.5 Conclusions

In this study, a novel MPC with closed-loop re-identification approach is developed that enables monitoring and updating the model used in the MPC using both past training data and current data. The proposed approach is described and compared against a representative offset-free MPC and shown to be able to provide improved closed-loop behavior through implementation on an example of a polymerization CSTR model subject to measurement noise.

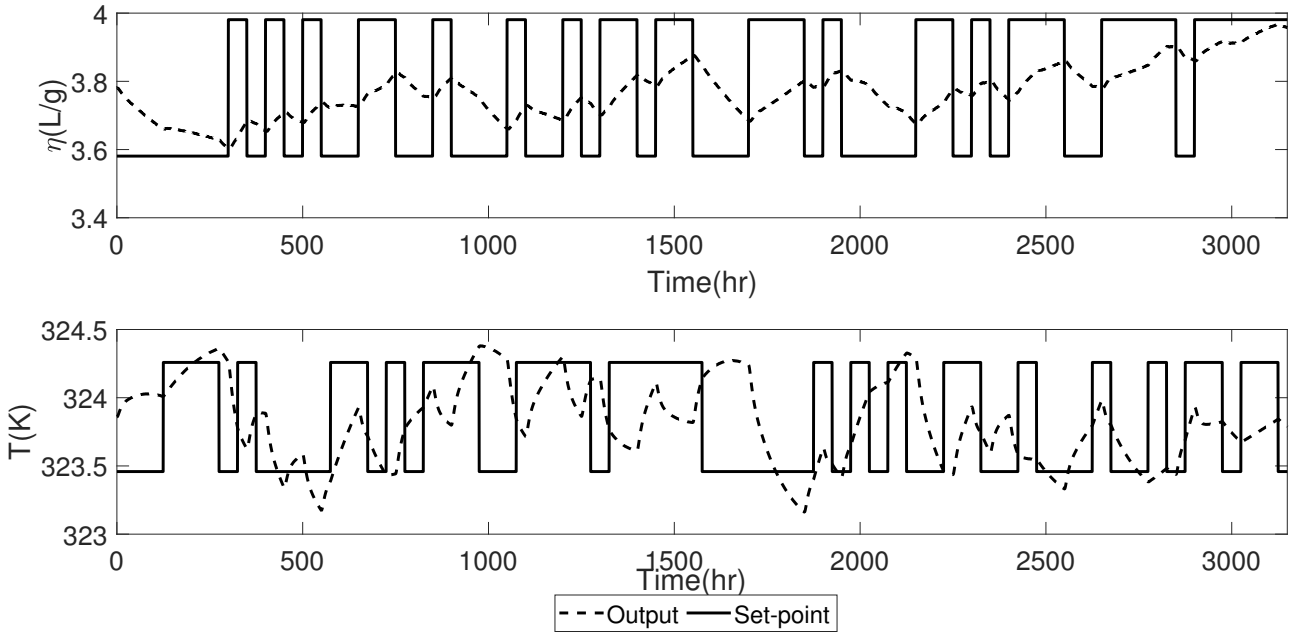

Figure 2.4: Model training data: measured outputs

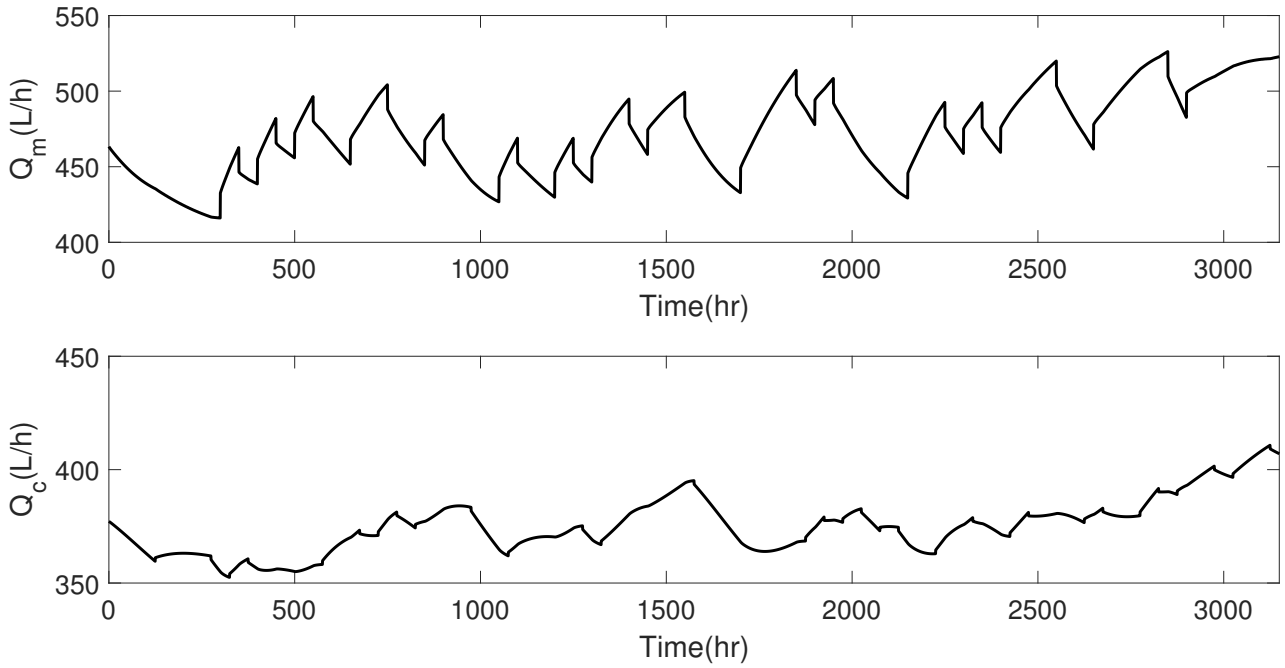

Figure 2.5: Model training data: manipulated inputs under PI controller

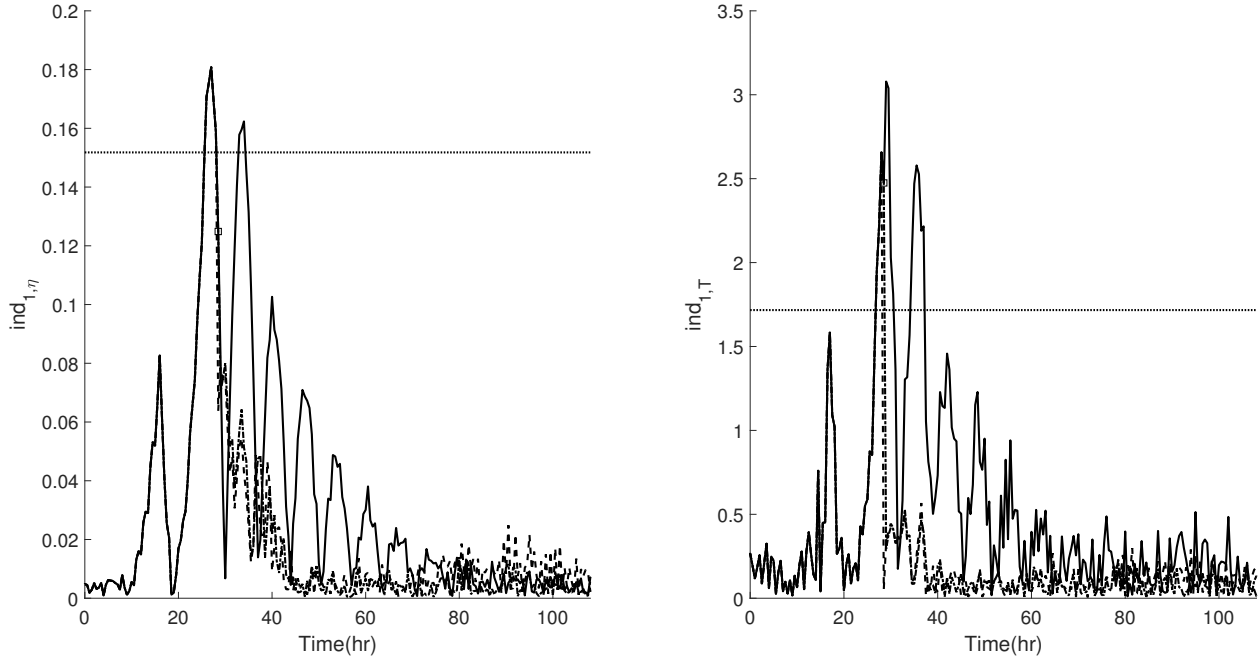

Figure 2.6: Model mean prediction error for the measured variables with re-identification (dash dotted line) and without re-identification (continuous line), the square indicates the re-identification trigger point and index threshold (dashed line)

Table 2.4: Controller parameters

Variable	Value
k_p	$\begin{bmatrix} 40.4 & 1.93 \end{bmatrix}$
k_I	$\begin{bmatrix} 1.96 & 0.258 \end{bmatrix}$
Q_y	$\begin{bmatrix} 50 & 0 \\ 0 & 5 \end{bmatrix}$
R_{du}	$\begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$
P	15
u_{min}	[71.6,78]
u_{max}	[870,670]
Δu_{min}	[-20,-20]
Δu_{max}	[20,20]
ϵ_{MPE}	[3.04,1.72]
ϵ_{EPPE}	[5.70,3.22]
Observer Poles	[0.06, 0.11, 0.16, 0.21, 0.26, 0.31]

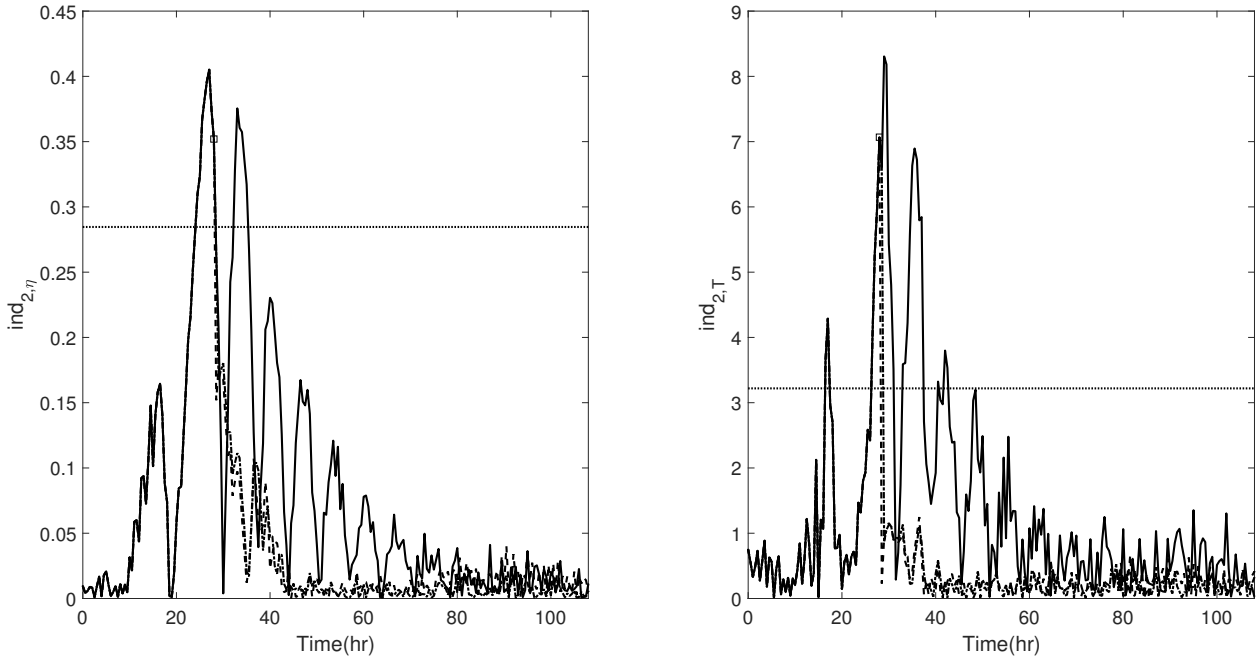

Figure 2.7: Model end-point mean prediction error for the measured variables with re-identification (dashed line) and without re-identification (continuous line), the square indicates the re-identification trigger point and index threshold (dashed line)

Table 2.5: LTI Model Parameters

Variable	Initial Model	Re-Identified Model with MPE
A	$\begin{bmatrix} 1.0083 & -0.0460 & 0.0784 & -0.0852 \\ 0.2095 & 1.1222 & 0.0035 & 0.0967 \\ -0.2993 & -0.0278 & 0.7651 & 0.0901 \\ 0.3255 & 0.1666 & 0.0774 & 0.9329 \end{bmatrix}$	$\begin{bmatrix} 0.9525 & 0.1490 & 0.0599 & 0.0903 \\ -0.2883 & 1.0229 & -0.4248 & 0.0352 \\ -0.2922 & 0.0962 & 0.5473 & 0.0619 \\ -0.2197 & 0.4309 & -0.2987 & 1.1981 \end{bmatrix}$
$B \times 10^3$	$\begin{bmatrix} 0.1544 & -0.0269 \\ -0.4656 & -0.5520 \\ -0.0060 & 1.5984 \\ -0.2566 & 0.8176 \end{bmatrix}$	$\begin{bmatrix} 0.0855 & 0.9243 \\ 0.1887 & 0.7452 \\ 0.1632 & 0.7534 \\ -0.0486 & 0.5613 \end{bmatrix}$
C	$\begin{bmatrix} -0.1569 & -0.2260 & 0.0429 & 0.2802 \\ 0.2115 & 0.1361 & 0.5661 & -0.2891 \end{bmatrix}$	$\begin{bmatrix} 0.0547 & 0.2300 & -0.5896 & -0.0713 \\ -0.0099 & -0.5014 & -0.3965 & 0.1270 \end{bmatrix}$
$Q \times 10^4$	$\begin{bmatrix} 0.0221 & -0.0575 & -0.0107 & -0.0309 \\ -0.0575 & 0.2147 & 0.0348 & 0.1155 \\ -0.0107 & 0.0348 & 0.0095 & 0.0189 \\ -0.0309 & 0.1155 & 0.0189 & 0.0647 \end{bmatrix}$	$\begin{bmatrix} 0.0433 & 0.0338 & 0.0087 & 0.0657 \\ 0.0338 & 0.0603 & 0.0225 & 0.0412 \\ 0.0087 & 0.0225 & 0.0105 & 0.0084 \\ 0.0657 & 0.0412 & 0.0084 & 0.1733 \end{bmatrix}$
$R \times 10^5$	$\begin{bmatrix} 0.1384 & 0.0667 \\ 0.0667 & 0.0619 \end{bmatrix}$	$\begin{bmatrix} 0.0245 & -0.0416 \\ -0.0416 & 0.4416 \end{bmatrix}$

Table 2.6: Comparison of nominal MPC and MPC with closed-loop re-identification (mean values)

Controller	Cost
Nominal MPC with $t_1 = 140$	136.5
Re-identification with MPE	110.8
Nominal MPC with $t_1 = 139$	148.1
Re-identification with $EPPE$	127.7

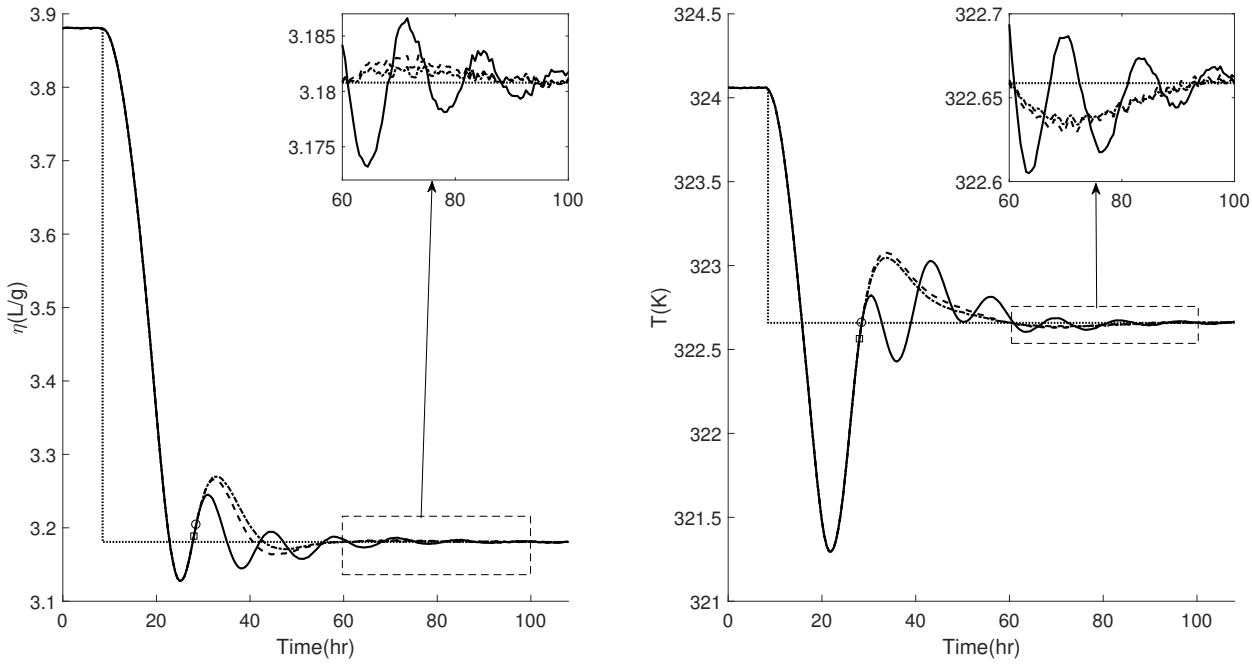

Figure 2.8: Comparison of the trajectories for the measured variables obtained from the proposed MPC with re-identification (dash dotted line with MPE, dashed line with EPPE) and conventional method (continuous line) and set-point (dotted line), the square and circle indicate the re-identification trigger points with MPE and EPPE indexes respectively

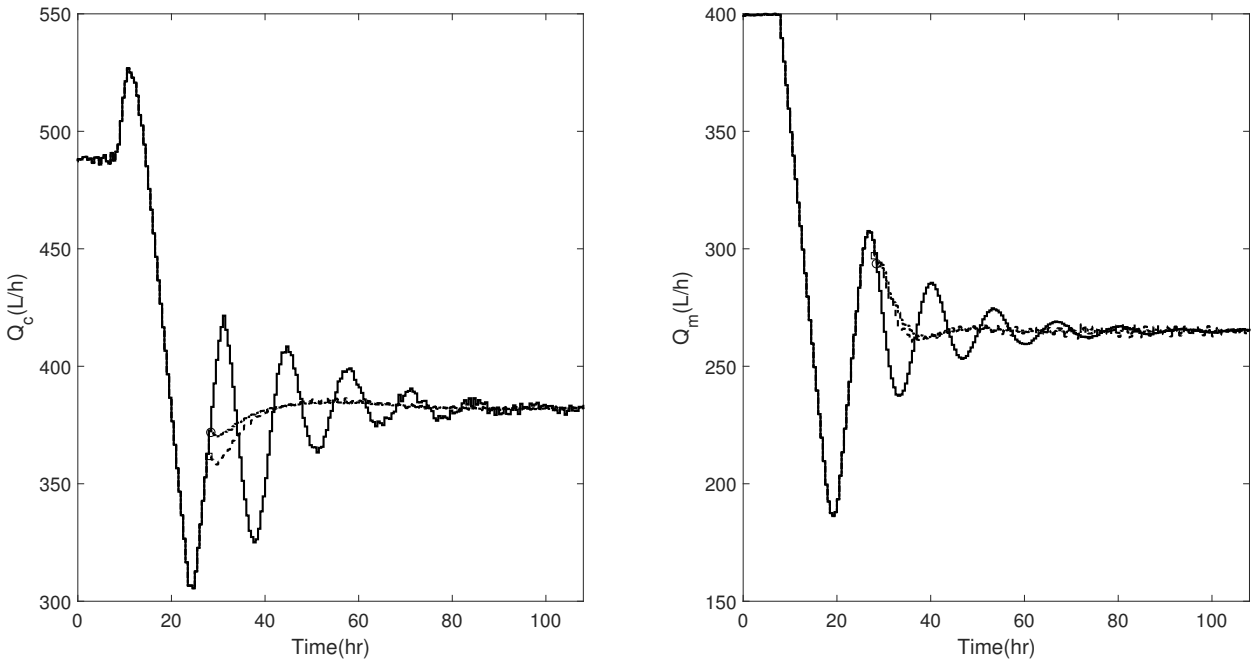

Figure 2.9: Closed-loop profiles of the manipulated variables obtained from the proposed MPC with re-identification (dash dotted line with MPE, dashed line with EPPE) and conventional method (continuous line), the square and circle indicate the re-identification trigger points with MPE and EPPE indexes respectively

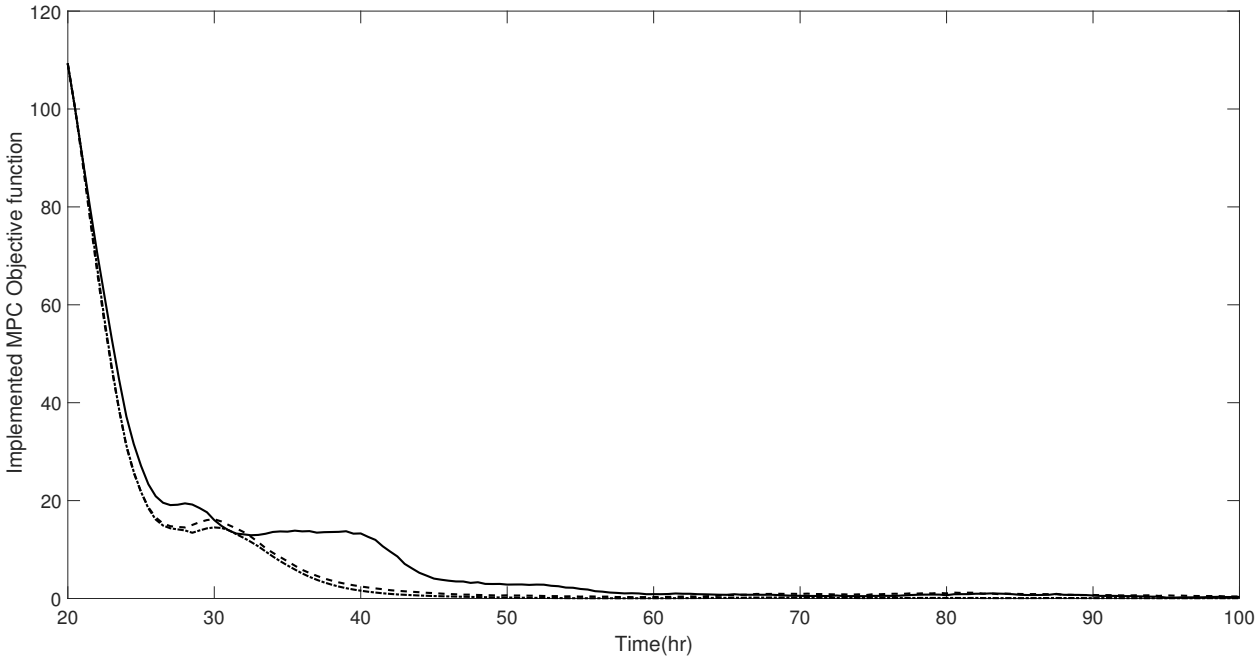

Figure 2.10: Closed-loop trajectories of the realized MPC objective function obtained from the proposed MPC with re-identification (dash dotted line with MPE, dashed line with EPPE) and conventional method (continuous line)

Chapter 3

MODEL PREDICTIVE CONTROL WITH RE-IDENTIFICATION OF BATCH PROCESSES[†]

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

[†]The results in this chapter have been submitted to: *Industrial & Engineering Chemistry Research*

using an electric arc furnace as a simulation test bed.

Keywords: Model Predictive Control, Batch Process Control, Subspace-Based Identification, Electric Arc Furnace

3.1 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 [9, 12, 16, 17, 37]. 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 [49, 63]. 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) [47]. In another approach, plant model mismatch is handled by integrating disturbances by so-called offset-free MPC, to overcome offset in set-point tracking [57, 70].

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 perfor-

mance monitoring. Many controller performance assessment approaches are based on comparing the controller performance with an ideal benchmark [11, 28, 65]. The prescribed remedy for poor performance is controller parameter retuning. In another direction [36] 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 built-in (IMPC). 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 [22] require identification at every time step. In later contributions [26, 59], 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 [36] would result in unnecessary re-identification. Furthermore, the results in [36] 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.

3.2 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.

3.2.1 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.

3.2.2 Subspace-Based Identification

In this section, a representative subspace based identification method is presented and reviewed [27, 67, 76]. 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:

$$x_{k+1} = Ax_k + Bu_k \tag{3.1}$$

$$y_k = Cx_k + Du_k \tag{3.2}$$

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:

$$U_p = U_{1|i_H} = \begin{bmatrix} u_1 & u_2 & \dots & u_j \\ u_2 & u_3 & \dots & u_{j+1} \\ \dots & \dots & \dots & \dots \\ u_{i_H} & u_{i_H+1} & \dots & u_{i_H+j-1} \end{bmatrix} \tag{3.3}$$

$$U_f = U_{i_H+1|2i_H} = \begin{bmatrix} u_{i_H+1} & u_{i_H+2} & \dots & u_{i_H+j} \\ u_{i_H+2} & u_{i_H+3} & \dots & u_{i_H+j+1} \\ \dots & \dots & \dots & \dots \\ u_{2i_H} & u_{2i_H+1} & \dots & u_{2i_H+j-1} \end{bmatrix} \tag{3.4}$$

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:

$$X_p = \begin{bmatrix} x_1 & x_2 & \dots & x_j \end{bmatrix} \tag{3.5}$$

$$X_f = \begin{bmatrix} x_{i_H+1} & x_{i_H+2} & \dots & x_{i_H+j} \end{bmatrix} \tag{3.6}$$

furthermore, the following variables are used in the approach:

$$\Psi_p = \begin{bmatrix} Y_p \\ U_p \end{bmatrix} \tag{3.7}$$

$$\Psi_f = \begin{bmatrix} Y_f \\ U_f \end{bmatrix} \tag{3.8}$$

By recursive substitution into the state space model equations Eqs. (3.1,3.2), it is straightforward to show:

$$Y_f = \Gamma_{i_H} X_f + \Phi^d_{i_H} U_f \tag{3.9}$$

$$Y_p = \Gamma_{i_H} X_p + \Phi^d_{i_H} U_p \tag{3.10}$$

$$X_f = A^{i_H} X_p + \Delta^d_{i_H} U_p \tag{3.11}$$

where:

$$\Gamma_{i_H} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i_H-1} \end{bmatrix} \tag{3.12}$$

$$\Phi_{i_H}^{d} = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CA^{i_H-2}B & CA^{i_H-3}B & CA^{i_H-4}B & \dots & D \end{bmatrix} \tag{3.13}$$

$$\Delta_{i_H}^{d} = \begin{bmatrix} A^{i_H-1}B & A^{i_H-2}B & \dots & AB & B \end{bmatrix} \tag{3.14}$$

Solving for X_f in Eq. 3.9 yields

$$X_f = \begin{bmatrix} \Gamma_{i_H}^{\dagger} & -\Gamma_{i_H}^{\dagger}\Phi_{i_H} \end{bmatrix} \begin{bmatrix} Y_f \\ U_f \end{bmatrix} \tag{3.15}$$

where $\Gamma_{i_H}^{\dagger}$ denotes the pseudo-inverse of Γ_i. It can be concluded from Eq. 3.15 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. 3.10 and by substituting into Eq. 3.11, we can write:

$$X_f = \begin{bmatrix} A^{i_H}\Gamma_{i_H}^{\dagger} & \Delta_{i_H} - A^{i_H}\Gamma_{i_H}^{\dagger}\Phi_{i_H} \end{bmatrix} \begin{bmatrix} Y_p \\ U_p \end{bmatrix} \tag{3.16}$$

From equation 3.16 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:

$$\text{span}\left(X_f\right) \coloneqq \text{row space}\left(\begin{bmatrix} Y_f \\ U_f \end{bmatrix}\right) \cap \text{row space}\left(\begin{bmatrix} Y_p \\ U_p \end{bmatrix}\right) \tag{3.17}$$

In [52], 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 [62].

3.2.2.1 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 [62], it was proposed to construct data-Hankel matrix for each batch b with input variable as the Hankel sub-matrix as:

$$U_p^{(b)} = U_{1|i_H}^{(b)} = \begin{bmatrix} u_1^{(b)} & u_2^{(b)} & \dots & u_{j^{(b)}}^{(b)} \\ \vdots & \vdots & \vdots & \vdots \\ u_{i_H}^{(b)} & u_{i_H+1}^{(b)} & \dots & u_{i_H+j^{(b)}-1}^{(b)} \end{bmatrix} \tag{3.18}$$

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

$$U_p = \begin{bmatrix} U_p^{(1)} & U_p^{(2)} & \dots & U_p^{(B)} \end{bmatrix} \tag{3.19}$$

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.

3.2.3 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:

$$\begin{aligned} &\tilde{x}_{k+1} = A\tilde{x}_k + B\tilde{u}_k, \\ &\tilde{y}_k = C\tilde{x}_k + D\tilde{u}_k, \\ &\tilde{u} \in \mathcal{U}, \qquad \tilde{x}(k) = \hat{x}_l, \end{aligned} \tag{3.20}$$

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 [34]. $\tilde{x}$ 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:

$$\begin{aligned} \hat{x}^-_k &= A\hat{x}_{k-1} + Bu_k \\ P^-_k &= AP_{k-1}A^T + Q \\ K_k &= P^-_k C^T \left(CP^-_k C^T + R\right)^{-1} \\ \hat{x}_k &= \hat{x}^-_k + K_k\left(y_k - C\hat{x}^-_k\right) \\ P_k &= \left(I - K_k C\right)P^-_k \end{aligned}$$

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 3.1. *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.*

3.2.4 Model monitoring and re-identification under continuous operation

We next review a recently proposed monitoring based MPC formulation [36]. 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 (3.1). In the schematic in Figure 3.1, the predicted output 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.

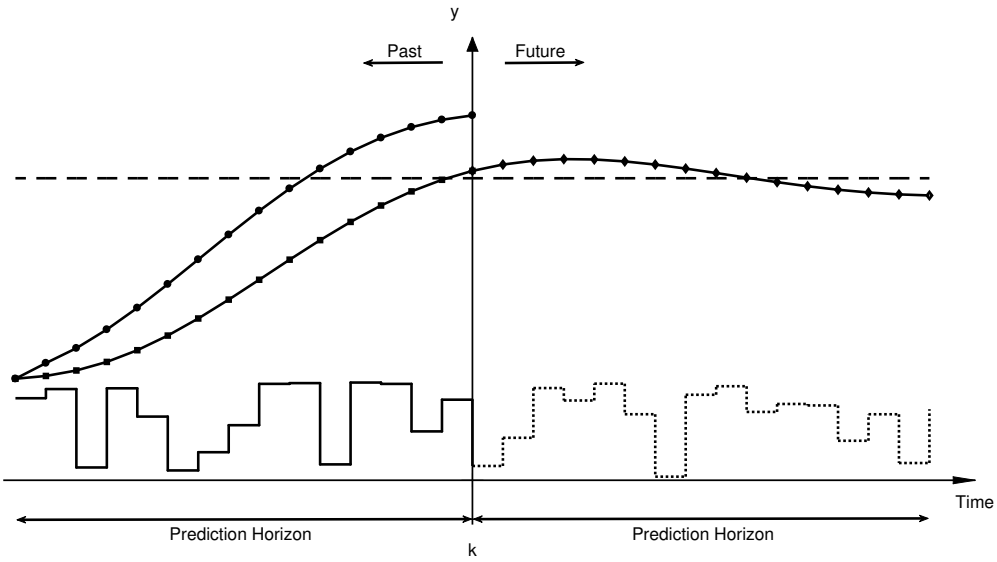

Figure 3.1: A schematic depicting an MPC implementation with re-identification for continuous operation −■− denotes the measured output, −●− denotes the past predicted output, −◆− denotes the Future Predicted Output, −− denotes Setpoint, − denotes Past Input, .. denotes Future Predicted Input)

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.

3.3 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.

3.3.1 Model performance index

In [36], 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:

$$MPE_k = \frac{1}{h} \sum_{j=k-h}^{k-1} (|y_j - \bar{y}_j|) \tag{3.21}$$

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:

$$MPE_{k,i} = max\{MPE_{k,i}^{(1)}, \ldots, MPE_{k,i}^{(B)}\}, \quad \text{For } i = 1, \ldots, n_y \tag{3.22}$$

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 3.2. *The definition of the index where it depends on the time into the batch renders the ability to set time varying threshold. This however, would not be readily implementable for batches with different duration. In such cases, an alignment variable could be used, or a threshold, which is the maximum over the training batches, could be used. Other alternatives include one where the error threshold is computed based on the proximity in state space in the training batches.*

3.3.2 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 (3.2). In order to evaluate model prediction performance at each sample time l, the estimated state vector

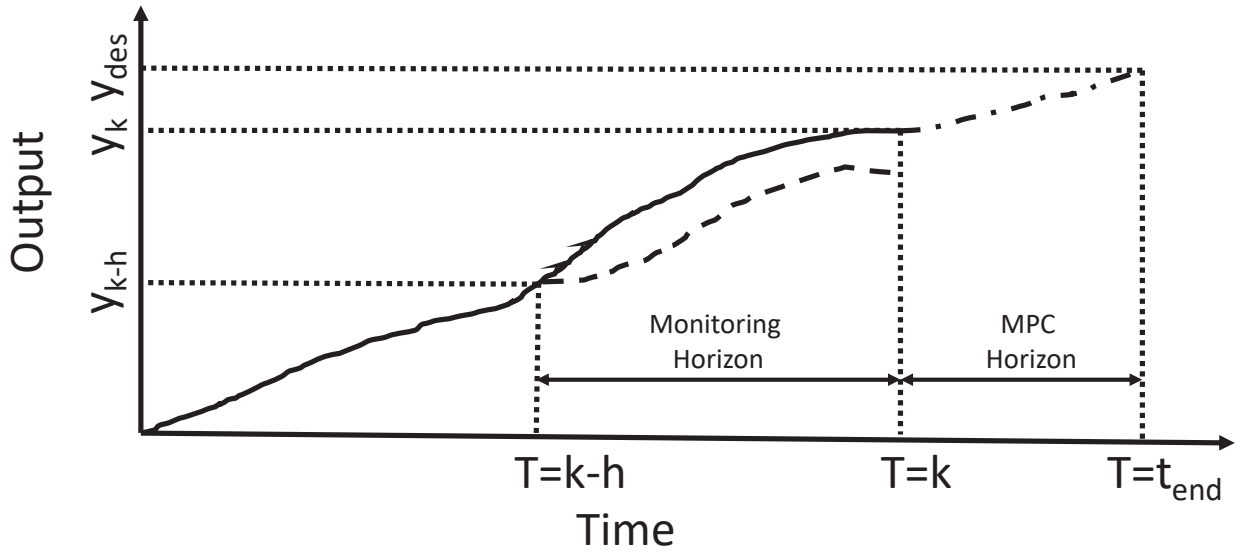

Figure 3.2: MPC with re-identification for batch systems ([Measured Output:continuous line], [Past Predicted Output: dashed line], [Future Predicted Output:dashed-dotted line])

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:

$$\begin{aligned} \bar{y}_j &= C\bar{x}_j + Du_j \\ \bar{x}_{j+1} &= A\bar{x}_j + Bu_j \\ \text{where } j &= l-h,\ldots,l-1 \\ \bar{x}_l &= \hat{x}_{l-h} \end{aligned} \tag{3.23}$$

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:

$$|y_{i,k} - \hat{y}_k| < \epsilon_i^* \quad \text{For } i = 1, \dots, n_y \tag{3.24}$$

where ϵ^* denotes the vector of acceptable output prediction error, and k^o represents the smallest sample time at which condition in equation 3.24 is satisfied.

$$k^o = inf\{k||y_{i,k} - \hat{y}_k| < \epsilon_i^* \quad \text{For } i = 1, \dots, n_y\} \tag{3.25}$$

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 (3.18)). Therefor, k^* is determined as follows:

$$k^* = h + max\{k^o, i_H\} \tag{3.26}$$

Remark 3.3. *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.*

3.3.3 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:

$$U_p = \begin{bmatrix} U_p^{(1)} & U_p^{(2)} & \dots & U_p^{(B)} & U_p^{(B)} & \dots & U_p^{(B)} \end{bmatrix} \tag{3.27}$$

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.

Remark 3.4. *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 3.5. *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 consists 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 [36]. 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).

3.3.4 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 3.6 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 3.7 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 3.6. *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 3.7. *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.

3.4 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.

3.4.1 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 [62], and are omitted here for brevity). A list of the process output variables are given in Table 3.1, and the inputs are listed in Table 3.2. 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.

Table 3.1: 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_{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_{CO_2}	Mass Fraction Carbon Dioxide in Gas	kg/kg
y_{10}	x_{N_2}	Mass Fraction Nitrogen in Gas	kg/kg

Table 3.2: 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_{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

3.4.2 Electric arc furnace model identification

As a test bed, a first principles EAF [8] 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 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 (3.3 & 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.

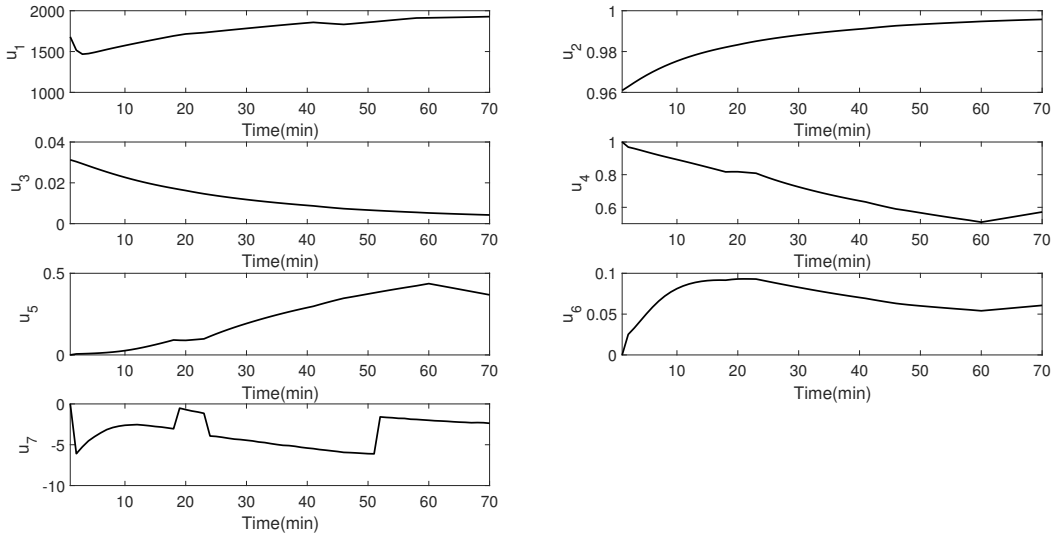

Figure 3.3: Input profiles for the validation batch

3.4.3 Model predictive control of the electric arc furnace

The proposed MPC with monitoring and identification was implemented on new batches. 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 Section 3.3.4, re-identification occurs at sample times $28, 34,\ 40,\ 46$ and 52, for both the MPC

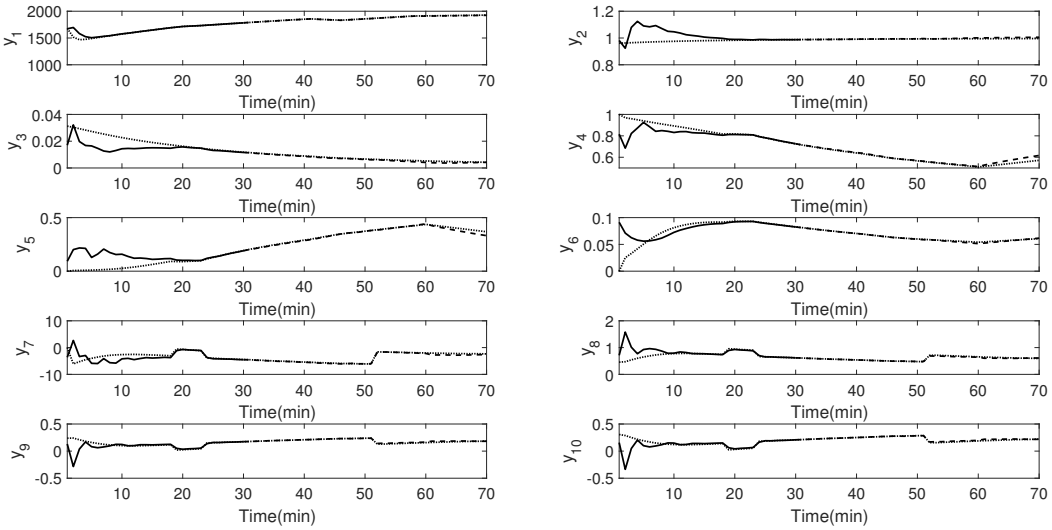

Figure 3.4: Validation batch: measured outputs (continuous line) and predicted output (dashed line)

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 (3.5-3.7). the prosed 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 (3.4). More importantly, the improved predictive ability of the re-identified models can be seen from the lower values of the monitoring index in Figure 3.7.

Remark 3.8. *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 3.3: 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 3.4: 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

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

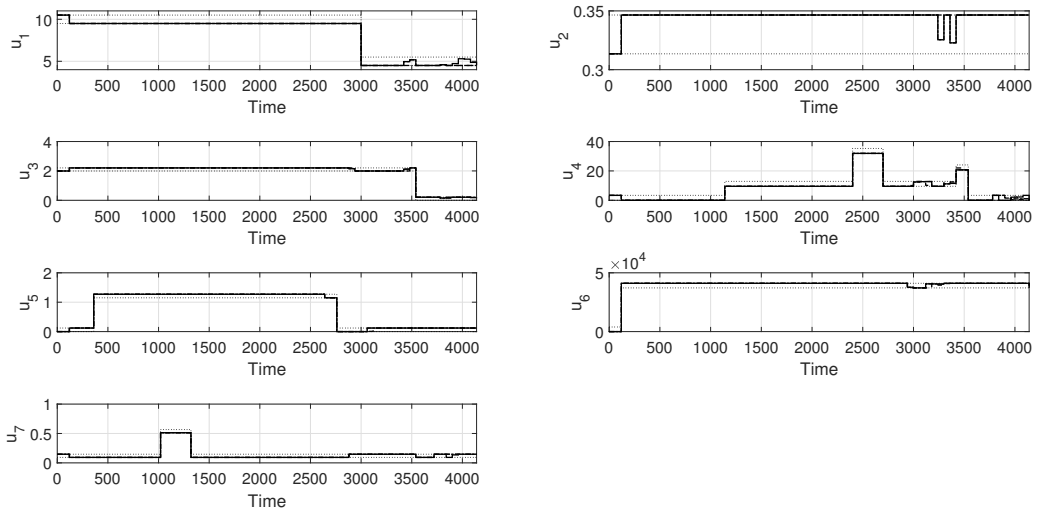

Figure 3.5: Closed-loop profiles of the input variables obtained from the proposed MPC and nominal MPC (Nominal MPC: dashed line, MPC with re-identification with one repetition : continuous line, MPC with re-identification with ten repetitions: dash-dotted line and lower and upper bounds on the inputs: dotted lines)

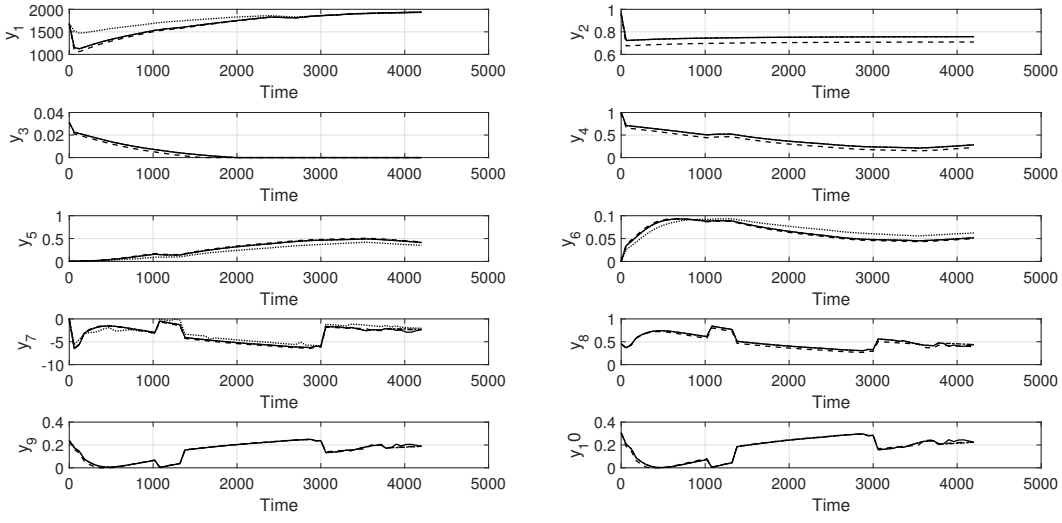

Figure 3.6: Comparison of the trajectories output variables obtained from the proposed MPC with re-identification and nominal MPC (Nominal MPC: dashed line, MPC with re-identification with one repetition : continuous line, MPC with re-identification with ten repetitions : dash-dotted line) and set-point for the controlled outputs: dotted line

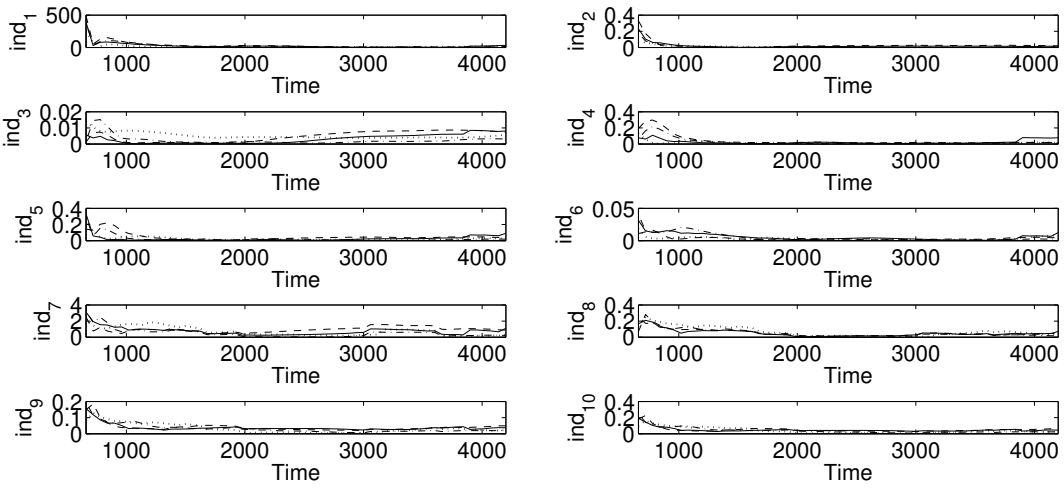

Figure 3.7: Comparison of the index trajectories obtained from the proposed MPC with re-identification and nominal MPC (Nominal MPC: dashed line, MPC with re-identification with one repetition : continuous line, MPC with re-identification with ten repetitions : dash-dotted line and threshold: dotted lines)

Chapter 4

DATA DRIVEN ECONOMIC MODEL PREDICTIVE CONTROL[†]

4.1 Introduction

Control systems designed to manage chemical process operations often faces numerous challenges such as inherent nonlinearity, process constraints and uncertainty. Model predictive control (MPC) is a well established control method that can handle these challenges. In MPC, the control action is computed by solving an open-loop optimal control problem at each sampling instance over a time horizon, subject to the model that captures the dynamic response of the plant, and constraints [63]. In early MPC designs, the objective function was often utilized as a parameter to ensure closed-loop stability. In subsequent contributions, Lyapunov-based MPC was proposed where feasibility and

[†]The results in this chapter have been published in[35]:

- M. Kheradmandi and P. Mhaskar. Data driven economic model predictive control. *Mathematics*, 6(4), p.51, 2018.

stability from a well characterized region was built into the MPC [46, 50].

With increasing recognition (and ability) of MPC designs to focus on economic objectives, the notion of Economic MPC (EMPC) was developed for linear and nonlinear systems [4, 7, 40], and several important issues (such as input rate-of-change constraint and uncertainty) addressed. The key idea with the EMPC designs is the fact that the controller is directly given the economic objective to work with, and the controller internally determines the process operation (including, if needed, a set point) [53].

Most of the existing MPC formulations, economic or otherwise, have been illustrated using first principles models. With growing availability of data, there exists the possibility of enhancing MPC implementation for situations where a first principles model may not be available, and simple 'step-test', transfer-function based model identification approaches may not suffice. One of the widely utilized approaches in the general direction of model identification are latent variable methods, where the correlation between subsequent measurements is used to model and predict the process evolution [23, 42]. In another direction, subspace-based system identification methods have been adapted for the purpose of model identification, where state-space model from measured data are identified using projection methods [24, 27, 38]. To handle the resultant plant model mismatch with data-driven model based approaches, monitoring of the model validity becomes especially important.

One approach to monitor the process is to focus on control performance [65], where the control performance is monitored and compared against a benchmark control design. To focus more explicitly on the model behavior, in a recent result [33] an adaptive data-driven MPC was proposed to evaluate model prediction performance and trigger model identification in case of poor model prediction. In another direction, an EMPC using empirical model was proposed [1]. The approach relies on a linearization approach, resulting in closed-loop stability guarantees for regions where the plant-model mismatch is sufficiently small, and illustrate results on stabilization around nom-

inally stable equilibrium points. In summary, data driven MPC or EMPC approaches, that utilize appropriate modeling techniques to identify data from closed-loop tests to handle operation around nominally unstable equilibrium points remain addressed.

Motivated by the above considerations, in this work we address the problem of data driven model based predictive control at an unstable equilibrium point. In order to identify a model around an unstable equilibrium point, the system is perturbed under closed-loop operation. Having identified a model, a Lyapunov-based MPC is designed to achieve local and practical stability. The Lyapunov-based design is then used as the basis for a data driven Lyapunov-based EMPC design to achieve economical goals while ensuring boundedness. The rest of the manuscript is organized as follows: First, the general mathematical description for the systems considered in this work, and a representative formulation for Lyapunov-based model predictive control are presented. Then the proposed approach for closed-loop model identification is explained. Subsequently, a Lyapunov-based MPC is designed, and illustrated through a simulation example. Finally, an economic MPC is designed to consider economical objectives. The efficacy of the proposed method is illustrated through implementation on a nonlinear continuous stirred-tank reactor (CSTR) with input rate of change constraints. Finally, concluding remarks are presented.

4.2 Preliminaries

This section presents a brief description of the general class of processes that are considered in this manuscript, followed by closed-loop subspace identification and Lyapunov based MPC formulation.

4.2.1 System Description

We consider a multi-input multi-output (MIMO) controllable systems where $u \in \mathbb{R}^{n_u}$ denotes the vector of constrained manipulated variables, 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, and $y \in \mathbb{R}^{n_y}$ denotes the vector of measured output variables. In keeping with the discrete implementation of MPC, 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. The central problem that the present manuscript addresses is that of designing a data driven modeling and control design for economic MPC.

4.2.2 System Identification

In this section, a brief review of a conventional subspace-based state space system identification methods is presented [29, 33, 60]. These methods are used to identify the system matrices for a discrete-time linear time invariant (LTI) system of the following form:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{4.1}$$

$$y_k = Cx_k + Du_k + v_k \tag{4.2}$$

where $x \in \mathbb{R}^{n_x}$ and $y \in \mathbb{R}^{n_y}$ denote the vectors of state variables and measured outputs, and $w \in \mathbb{R}^{n_x}$ and $v \in \mathbb{R}^{n_y}$ are zero mean, white vectors of process noise and measurement noise with the following

covariance matrices:

$$E[\begin{pmatrix} w_i \\ v_j \end{pmatrix}\begin{pmatrix} w_i^T & v_j^T \end{pmatrix}] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}\delta_{ij} \tag{4.3}$$

where $Q \in \mathbb{R}^{n_x \times n_x}$, $S \in \mathbb{R}^{n_x \times n_y}$ and $R \in \mathbb{R}^{n_y \times n_y}$ are covariance matrices, and, δ_{ij} is the Kronecker delta function. The subspace-based system identification techniques utilize Hankel matrices constructed by stacking the output measurements and manipulated variables as follows:

$$U_{1|i} = \begin{bmatrix} u_1 & u_2 & \dots & u_j \\ u_2 & u_3 & \dots & u_{j+1} \\ \dots & \dots & \dots & \dots \\ u_i & u_{i+1} & \dots & u_{i+j-1} \end{bmatrix} \tag{4.4}$$

where i is a user-specified parameter that limits the maximum order of the system (n), and, j is determined by the number of sample times of data. By using Eq. 4.4, the past and future Hankel matrices for input and output are defined:

$$U_p = U_{1|i}, \quad U_f = U_{1|i}, \quad Y_p = Y_{1|i}, \quad Y_f = Y_{1|i} \tag{4.5}$$

Similar block-Hankel matrices are made for process and measurement noises $V_p, V_f \in \mathbb{R}^{in_y \times j}$ and $W_p, W_f \in \mathbb{R}^{in_x \times j}$ are defined in the similar way. The state sequences are defined as follows:

$$X_p = \begin{bmatrix} x_1 & x_2 & \dots & x_j \end{bmatrix} \tag{4.6}$$

$$X_f = \begin{bmatrix} x_{i+1} & x_{i+2} & \dots & x_{i+j} \end{bmatrix} \tag{4.7}$$

furthermore these matrices are used in the algorithm:

$$\Psi_p = \begin{bmatrix} Y_p \\ U_p \end{bmatrix}, \quad \Psi_f = \begin{bmatrix} Y_f \\ U_f \end{bmatrix}, \quad \Psi_{pr} = \begin{bmatrix} R_f \\ \Psi_p \end{bmatrix} \tag{4.8}$$

By recursive substitution into the state space model equations Eqs. (4.1,4.2), it is straightforward to show:

$$Y_f = \Gamma_i X_f + \Phi_i^d U_f + \Phi_i^s W_f + V_f \tag{4.9}$$

$$Y_p = \Gamma_i X_p + \Phi_i^d U_p + \Phi_i^s W_p + V_p \tag{4.10}$$

$$X_f = A^i X_p + \Delta_i^d U_p + \Delta_i^s W_p \tag{4.11}$$

where:

$$\Gamma_i = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{bmatrix}, \quad \Phi_i^d = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & \dots & D \end{bmatrix} \tag{4.12}$$

$$\Phi_i^s = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ C & 0 & 0 & \dots & 0 & 0 \\ CA & C & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & 0 & 0 \\ CA^{i-2} & CA^{i-3} & CA^{i-4} & \dots & C & 0 \end{bmatrix} \tag{4.13}$$

$$\Delta_i^d = \begin{bmatrix} A^{i-1}B & A^{i-2}B & \dots & AB & B \end{bmatrix}, \quad \Delta_i^s = \begin{bmatrix} A^{i-1} & A^{i-2} & \dots & A & I \end{bmatrix} \tag{4.14}$$

Eq. (4.9) can be rewritten in the following form to have the input and output data at the LHS of the equation[71]:

$$\begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \begin{bmatrix} Y_f \\ U_f \end{bmatrix} = \Gamma_i X_f + \Phi_i^s W_f + V_f \tag{4.15}$$

In open loop identification methods, in the next step, by orthogonal projecting of Eq. (4.15) onto Ψ_p:

$$\begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \Psi_f / \Psi_p = \Gamma_i X_f / \Psi_p \tag{4.16}$$

Note that, the last two terms in RHS of Eq. (4.15) are eliminated since the noise terms are independent, or othogonal to the future inputs. Eq (4.16) indicates that:

$$Column\_Space(W_f/W_p) = Column\_Space((\Gamma_i{}^{\perp^T} \begin{bmatrix} I & -H_i^d \end{bmatrix})^T) \tag{4.17}$$

Therefore Γ_i and H_i^d can be calculated using Eq (4.17) by decomposition methods. These can in turn be utilized to determine the system matrices (some of these details are deferred to Section 4.3.1). For further discussion on system matrix extraction, the readers are referred to references [29, 60].

4.2.3 Lyapunov-Based MPC

The Lyapunov-based MPC (LMPC) for linear system has the following form:

$$\min_{\tilde{u}_k,\ldots,\tilde{u}_{k+P}} \sum_{j=1}^{N_y} ||\tilde{y}_{k+j} - y_{k+j}^{\text{SP}}||_{Q_y}^2 + \sum_{j=1}^{N_u} ||\tilde{u}_{k+j} - \tilde{u}_{k+j-1}||_{R_{du}}^2 \tag{4.18}$$

subject to: (4.19)

$$\tilde{x}_{k+1} = A\tilde{x}_k + B\tilde{u}_k \tag{4.20}$$

$$\tilde{y}_k = C\tilde{x}_k + D\tilde{u}_k \tag{4.21}$$

$$\tilde{u} \in \mathcal{U}, \qquad \Delta\tilde{u} \in \mathcal{U}_\delta, \qquad \tilde{x}(k) = \hat{x}_l \tag{4.22}$$

$$V(\tilde{x}_{k+1}) \leq \alpha V(\tilde{x}_k) \; \forall \; V(\tilde{x}_k) > \epsilon^* \tag{4.23}$$

$$V(\tilde{x}_{k+1}) \leq \epsilon^* \; \forall \; V(\tilde{x}_k) \leq \epsilon^* \tag{4.24}$$

where $\tilde{x}_{k+j}$, $\tilde{y}_{k+j}$, y^{SP}_{k+j} and $\tilde{u}_{k+j}$ denote predicted state and output, output set-point and calculated manipulated input variables j time steps ahead computed at time step k, and $\hat{x}_l$ is the current estimation of state, and $< 0\alpha < 1$ is a user defined parameter. The operator $||.||^2_Q$ denotes the weighted Euclidean norm defined for an arbitrary vector x and weighting matrix W as $||x||^2_W = x^T W x$. Further, $Q_y > 0$ and $R_{du} \geq 0$ denote the positive definite and positive semi-definite weighting matrices for penalizing deviations in the output predictions and for the rate of change of the manipulated inputs, respectively. Moreover, N_y and N_u denote the prediction and control horizons, respectively, and the input rate of change, given by $\Delta\tilde{u}_{k+j} = \tilde{u}_{k+j} - \tilde{u}_{k+j-1}$, takes values in a nonempty convex subset $\mathcal{U}_\delta \subset \mathbb{R}^m$, where $\mathcal{U}_\delta = \left\{\Delta u \in \mathbb{R}^{n_u} \mid \Delta u_{\min} \leq \Delta u \leq \Delta u_{\max}\right\}$. Note finally, that while the system dynamics are described in continuous time, the objective function and constraints are defined in discrete time to be consistent with the discrete implementation of the control action.

Eqs. 4.23 and 4.24 are representatives of Lyapunov-based stability constraint [48, 51], where $V(x_k)$ is a suitable control Lyapunov function, and $\alpha, \epsilon^* > 0$ are user-specified parameter. In the presented formulation, $\epsilon^* > 0$ enables practical stabilization to account for the discrete nature of the control implementation.

Remark 4.1. *Existing Lypunov-based MPC approaches exploit the fact that the feasibilty (and stability) region can be pre-determined. The feasibility region, among other things, depends on*

the choice of the parameter α, the requested decay factor in the value of the Lyapunov function at each time step. If (reasonably) good first principles modesl are available, then these features of the MPC formulation provide excellent confidence over the operating region under closed-loop. In contrast, in the presence of significant plant-model mismatch (as is possibly the case with data driven models), the imposition of such decay constraints could result in unnecessary infeasibility issues. In designing the LMPC formulation with a data driven model, this possible lack of feasibility must be accounted for (as is done in Section 4.3.2).

4.3 Integrating Lyapunov-based MPC with Data Driven Models

In this section, we first utilize an identification approach necessary to identify good models for operation around an unstable equilibrium point. The data driven Lyapunov- based MPC design is presented next.

4.3.1 Closed-loop Model Identification

Note that when interested in identifying the system around an unstable equilibrium point, open-loop data would not suffice. To begin with, nominal open-loop operation around an unstable equilibrium point is not possible. If the nominal operation is under closed-loop, but the loop is opened to perform step tests, the system would move to the stable equilibrium point corresponding to the new input value, thereby not providing dynamic information around the desired operating point. The training data, therefore, has to be obtained using closed-loop step tests, and an appropriate closed-loop model identification method employed. Such a method is next described.

In employing closed-loop data, note that the assumption of future inputs being independent of future disturbances no longer holds, and if not recognized, can cause biased results in system identification

[29]. In order to handle this issue, the closed-loop identification approach in the projection utilizes a different variable Ψ_{pr} instead of Ψ_p. The new instrument variable, that satisfies the independence requirement, is used to project both side of the Equation 4.15 and the result is used to determine LTI model matrices. For further details, refer to [29, 33, 61].

By projecting Eq. (4.15) onto Ψ_{pr} we get:

$$\begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \Psi_f / \Psi_{pr} = \Gamma_i X_f / \Psi_{pr} + \Phi_i^s W_f / \Psi_{pr} + V_f / \Psi_{pr} \tag{4.25}$$

Since the future process and measurement noises are independent of the past input/output and future setpoint in Eq. (4.25), the noise terms cancel, resulting in:

$$\begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \Psi_f / \Psi_{pr} = \Gamma_i X_f / \Psi_{pr} \tag{4.26}$$

By multiplying Eq. (4.26) by the extended orthogonal observability $\Gamma_i^{\perp}$, the state term is eliminated:

$$(\Gamma_i^{\perp})^T \begin{bmatrix} I & -\Phi_i^d \end{bmatrix} \Psi_f / \Psi_{pr} = 0 \tag{4.27}$$

Therefore the column space of Ψ_f / Ψ_{pr} is orthogonal to the row space of $\begin{bmatrix} (\Gamma_i^{\perp})^T & -(\Gamma_i^{\perp})^T \Phi_i^d \end{bmatrix}$. By performing singular value decomposition (SVD) of Ψ_f / Ψ_{pr}:

$$\Psi_f / \Psi_{pr} = U \Sigma V = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {V_1}^T \\ {V_2}^T \end{bmatrix} \tag{4.28}$$

where Σ_1 contains dominant singular values of Ψ_f / Ψ_{pr} and, theoretically it has the order $n_u i + n$ [29, 61].

Therefore the order of the system can be determined by the number of the dominant singular values

of the Ψ_f/Ψ_{pr} [71]. The orthogonal column space of Ψ_f/Ψ_{pr} is U_2M, where $M \in \mathbb{R}^{(n_y-n)i\times(n_y-n)i}$ is any constant nonsingular matrix and is typically chosen as an identity matrix[29, 61]. One approach to determine the LTI model is as follows [29]:

$$(\begin{bmatrix}\Gamma_i^{\perp} & -\Gamma_i^{\perp}\Phi_i^d\end{bmatrix})^T = U_2M \tag{4.29}$$

From Eq.(4.29), Γ_i and Φ_i^d can be estimated.

$$\begin{bmatrix}\Gamma_i{}^{\perp} \\ -(\Phi_i^d)^T\Gamma_i{}^{\perp}\end{bmatrix} = U_2 \tag{4.30}$$

which results in (using MATLAB matrix index notation):

$$\begin{cases}\hat{\Gamma}_i = U_2(1:n_yi,:)^{\perp} \\ \hat{\Phi}_i^d = -(U_2(1:n_yi,:)^T)^{\dagger}U_2(n_yi+1:end,:)^T\end{cases} \tag{4.31}$$

The past state sequence can be calculated as follows:

$$\hat{X}_i = \hat{\Gamma}_i^{\dagger}\begin{bmatrix}I & -\hat{\Phi}_i^d\end{bmatrix}\Psi_f/\Psi_{pr} \tag{4.32}$$

The future state sequence can be calculated by changing data Hankel matrices as follows [29]:

$$R_f = R_{i+2|2i} \tag{4.33}$$

$$U_p = U_{1|i+1} \tag{4.34}$$

$$Y_p = Y_{1|i+1} \tag{4.35}$$

$$U_f = U_{i+2|2i} \tag{4.36}$$

$$Y_f = Y_{i+2|2i} \tag{4.37}$$

$$\Rightarrow \hat{X}_{i+1} = \hat{\underline{\Gamma}}_i^{\dagger} \begin{bmatrix} I & -\underline{\hat{H}}_i^d \end{bmatrix} \Psi_f / \Psi_{pr} \tag{4.38}$$

where $\hat{\underline{\Gamma}}_i$ is obtained by eliminating the last n_y rows of Γ_i, and $\underline{H}_i^d$ is obtained by eliminating the last n_y rows and the last n_u columns of H_i^d. Then the model matrices can be estimated using least square:

$$\begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} W_{i|i} \\ V_{i|i} \end{bmatrix} \tag{4.39}$$

Note that the difference between the proposed method in [29] and described method is that in order to ensure that observer is stable (eigenvalues of $A - KC$ are inside unit circle), instead of innovation form of LTI model, Equations (4.1,4.2) are used[33] to derive extended state space equations. The system matrices can be calculated as follows:

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix}^{\dagger} \tag{4.40}$$

With the proposed approach, process and measurement noise Hankel matrices can be calculated as the residual of the least square solution of Eq. 4.39:

$$\begin{bmatrix} \hat{W}_{i|i} \\ \hat{V}_{i|i} \end{bmatrix} = \begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix} \tag{4.41}$$

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

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = E\left(\begin{bmatrix} \hat{W}_{i|i} \\ \hat{V}_{i|i} \end{bmatrix} \begin{bmatrix} \hat{W}_{i|i}^T & \hat{V}_{i|i}^T \end{bmatrix} \right) \tag{4.42}$$

Model identification using closed-loop data has a positive impact on the predictive capability of the model (see the simulation section for a comparison with a model identified using open-loop data).

4.3.2 Control Design and Implementation

Having identified an LTI model for the system (with its associated states), the MPC implementation first requires a determination of the state estimates. To this end, an appropriate state estimator needs to be utilized. In the present manuscript, a Luenberger observer is utilized for the purpose of illustration. Thus, at the time of control implementation, state estimates $\hat{x}_k$ are generated as follows:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - C\hat{x}_k) \tag{4.43}$$

where L is the observer gain and is computed using pole placement method, and y_k is the vector of measured variables (in deviation form, from the set point).

In order to stabilize the system at unstable equilibrium point, a Lyapunov-based MPC is designed. The control calculation is achieved using a two tier approach (to decouple the problem of stability enforcement and objective function tuning). The first layer calculates the minimum value of Lyapunov function that can be reached subject the constraints. This tier is formulated as follows:

$$\begin{aligned} &V_{min} = \min_{\tilde{u}_k^1}(V(\tilde{x}_{k+1})) \\ &\text{subject to:} \\ &\qquad \tilde{x}_{k+1} = A\tilde{x}_k + B\tilde{u}_k^1 \\ &\qquad \tilde{y}_k = C\tilde{x}_k + D\tilde{u}_k^1 \\ &\qquad \tilde{u}^1 \in \mathcal{U}, \qquad \Delta\tilde{u}^1 \in \mathcal{U}_\delta, \qquad \tilde{x}(k) = \hat{x}_l - x^{SP} \end{aligned} \tag{4.44}$$

where $\tilde{x}$, $\tilde{y}$ are predicted state and output and $\tilde{u}^1$ is the candidate input computed in the first tier. x^{SP} is underlying state setpoint (in deviation form from the nominal equilibrium point) which here is the desired unstable equilibrium point (and therefore zero in terms of deviation variables). For setpoint tracking this value can be calculated using target calculation method; readers are referred to [57] for further details.

Note that the first tier has a prediction horizon of 1, because the objective is to only compute the immediate control action that would minimize the value of the Lyapunov function at the next time step. V is chosen as a quadratic Lyapunov function with the following form:

$$V(\tilde{x}) = \tilde{x}^T P \tilde{x} \tag{4.45}$$

where P is a positive definite matrix computed by solving the Riccati equation with the LTI model matrices as follows:

$$A^T P A - P - A^T P B (B^T P B + R)^{-1} + Q = 0 \tag{4.46}$$

where $Q \in \mathbb{R}^{n_x \times n_x}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are positive definite matrices. Then in the second tier this minimum value is used as a constraint (upper bound for Lyapunov function value at the next time

step). The second tier is formulated as follows:

$$\min_{\tilde{u}^2_k,\dots,\tilde{u}^2_{k+N_p}} \sum_{j=1}^{N_y} ||\tilde{y}_{k+j} - \tilde{y}^{\text{SP}}_{k+j}||^2_{Q_y} + ||\tilde{u}^2_{k+j} - \tilde{u}^2_{k+j-1}||^2_{R_{du}}$$

subject to:

$$\begin{aligned}
&\tilde{x}_{k+1} = A\tilde{x}_k + B\tilde{u}_k \\
&\tilde{y}_k = C\tilde{x}_k + D\tilde{u}_k \\
&\tilde{u}^2 \in \mathcal{U}, \qquad \Delta\tilde{u}^2 \in \mathcal{U}_\delta, \qquad \tilde{x}(k) = \hat{x}_l \\
&V(\tilde{x}_{k+1}) \leq V_{min} \;\forall\; V(\tilde{x}_k) > \epsilon^* \\
&V(\tilde{x}_{k+1}) \leq \epsilon^* \;\forall\; V(\tilde{x}_k) \leq \epsilon^*
\end{aligned} \tag{4.47}$$

where N_p is the prediction horizon and $\tilde{u}^2$ denotes the control action computed by the second tier. In essence, in the second tier, the controller calculates a control action sequence that can take the process to the setpoint in an optimal fashion optimally while ensuring that system reaches the minimum achievable Lyapunov function value at the next time step. Note that in both the tiers, the input sequence is a decision variable in the optimization problem, but only the first value of the input sequence of the second tier is implemented on the process. The solution of the first tier, however, is used to ensure and generate a feasible initial guess for the second tier.

Remark 4.2. *Note that Tiers 1 and 2 are executed in series and at the same time, and the implementation does not require a time scale separation. The overall optimization is split into two tiers to guarantee feasibility of the optimization problem. In particular, the first tier computes an input move with the objective function only focusing on minimizing the Lyapunov function value at the next time step. Notice that the constraints in the first tier are such that the optimization problem is guaranteed to be feasible. With this feasible solution, the second tier is used to determine the input trajectory that achieves the best performance, while requiring the Lyapunov function to decay. Again, since the second Tier optimization problem uses the solution from Tire 1 to impose the stability constraint,*

feasibility of the second Tier optimization problem, and hence of the MPC optimization problem is guaranteed. In contrast, if one were to require the Lyapunov function to decay by an arbitrary chosen factor, determination of that factor in a way that guarantees feasibility of the optimization problem would be a non-trivial task.

Remark 4.3. *It is important to recognize that in the present formulation, feasibility of the optimization problem does not guarantee closed-loop stability. A superfluous (and incorrect) reason is as follows: the first tier computes the control action that minimizes the value of the Lyapunov function at the next step, but does not require that it be smaller than the previous time step, leading to potential destabilizing control action. The key point to realize here, however, is that if such a control action were to exist (that would lower the value of the Lyapunov function at the next time step), the optimization problem would determine that value by virtue of the Lypaunov function being the objective function, and lead to closed-loop stability. The reasons closed-loop stability may not be achieved are two: 1) the current state might be such that closed-loop stability is not achievable for the system dynamics and constraints, and 2) due to plant model mismatch, where the control action that causes the Lyapunov function to decay for the identified model does not do so for the system in question. The first reason points to a fundamental limitation due to the presence of input constraints, while the second is due to the lack of availability of the 'correct' system dynamics, and as such will be true in general for data driven MPC formulations.*

Remark 4.4. *In the current manuscript, we focus on the cases where a first principal model is not available. If a good first principles model was available, it could be utilized directly in a nonlinear MPC design, or linearized if one were to implement a linear MPC. In the case of linearization, the applicability would be limited by the region over which the linearization holds. In contrast, note that the model utilized in the present manuscript does not result from a linearization of a nonlinear model. Instead it is a linear model, possibly with higher number of states than the original nonlinear model, albeit identified, and applicable, over a 'larger' region of operation, compared to a linearized model.*

Remark 4.5. *To account for possible plant-model mismatch, model validity can be monitored with model monitoring methods [33], resulting in appropriately triggering re-identification in case of poor model prediction. In another direction, in line with control performance monitoring approaches, the Lyapunov function value could be utilized. Thus, unacceptable increases in Lyapunov function value could be utilized as a means of triggering re-identification.*

Remark 4.6. *As mentioned previously, in order to create a rich training data around unstable operating point, closed-loop data must be generated. In turn, since open-loop methods result in biased estimation [18, 41] in model identification, a suitable closed-loop identification method is utilized, and adapted to ensure that the model accurately captures the key dynamics.*

4.4 Simulation Results

We next illustrate the proposed approach using a nonlinear CSTR example [70]. To this end, consider a continuous stirred-tank reactor (CSTR) where a first-order, exothermic and irreversible reaction of the form $A \xrightarrow{k} B$ takes place. The mass and energy conservation laws results in the following mathematical model:

$$\begin{aligned} \dot{C}_A &= \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT_R}} C_A \\ \dot{T}_R &= \frac{F}{V}(T_{A0} - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 e^{\frac{-E}{RT_R}} C_A + \frac{Q}{\rho c_p V} \end{aligned} \tag{4.48}$$

The description of the process variables and the values of the system parameters are presented in Table 4.1. The control objective is stabilize the system at an unstable equilibrium point using inlet concentration C_{A_0}, and the rate of heat input, Q, while the manipulated inputs are constrained to be within the limits $|C_{A_0}| \leq 1 Kmol/m^3$ and $|Q| \leq 9 \times 10^3 KJ/min$, and the input rate is constrained as $|\Delta C_{A_0}| \leq 0.1 Kmol/m^3$ and $|\Delta Q| \leq 9 \times 200 KJ/min$. We assume that both of the states are measured. The system has an unstable equilibrium point at $C_A = 0.573 kmol/m^3$ and $T = 395.3K$.

The goal is to stabilize the system at this equilibrium point. To this end, first an LTI model is identified using closed-loop data, then an MPC is designed to stabilize the system at the unstable equilibrium point.

For system identification of the CSTR model, PI controllers (pairing C_A with $C_{A,in}$ and T with Q) are implemented on the process. In particular, pseudo-random binary signals are used as set-point for PI controllers. The identified LTI model order is selected as $n = 4$ and $i = 12$, to achieve the best fit in model prediction (using cross-validation). Model validation results under a different set of set-point changes from training data is presented in Figures (4.1 & 4.2). The identified system is unstable with absolute eigenvalues $\begin{bmatrix} 0.9311 & 0.9311 & 0.9998 & 1.0002 \end{bmatrix}$ which has an eigenvalue outside unit circle. The unstable nature of the identified model is consistent with the operation of the system around the unstable equilibrium point.

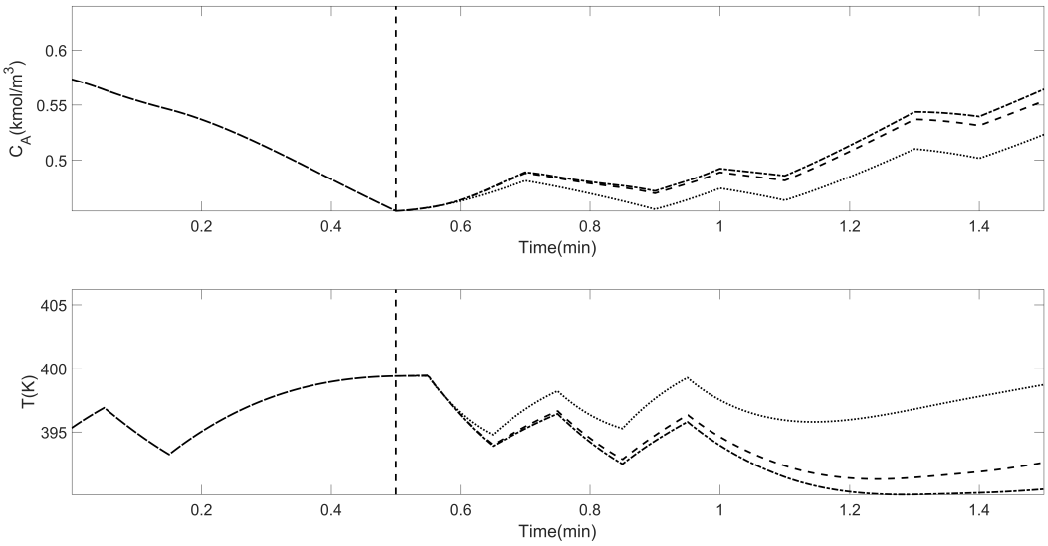

Figure 4.1: Data driven model validation results: measured outputs (dash-dotted line), state and output estimates using the LTI model from closed-loop data and identification (dashed line), state and output estimates using the LTI model from open-loop data and identification (dotted line), observer stopping point (vertical dashed line)

Table 4.1: Variable and parameter description and values for the CSTR example

Variable	Description	Unit	Value
$C_{A,S}$	Nominal Value of Concentration	$\frac{kmol}{m^3}$	0.573
$T_{R,S}$	Nominal Value of Reactor Temperature	K	395
F	Flow Rate	$\frac{m^3}{min}$	0.2
V	Volume of the Reactor	$\frac{m^3}{min}$	0.2
$C_{A0,S}$	Nominal Inlet Concentration	$\frac{kmol}{m^3}$	0.787
k_0	Pre-Exponential Constant	$-$	72×10^9
E	Activation Energy	$\frac{kJ}{mol}$	8.314×10^4
R	Ideal Gas Constant	$\frac{kJ}{kmol.K}$	8.314
T_{A0}	Inlet Temperature	K	352.6
ΔH	Enthalpy of the Reaction	$\frac{kJ}{kmol}$	4.78×10^4
ρ	Fluid Density	$\frac{kg}{m^3}$	10^3
c_p	Heat Capacity	$\frac{kj}{kg.K}$	0.239

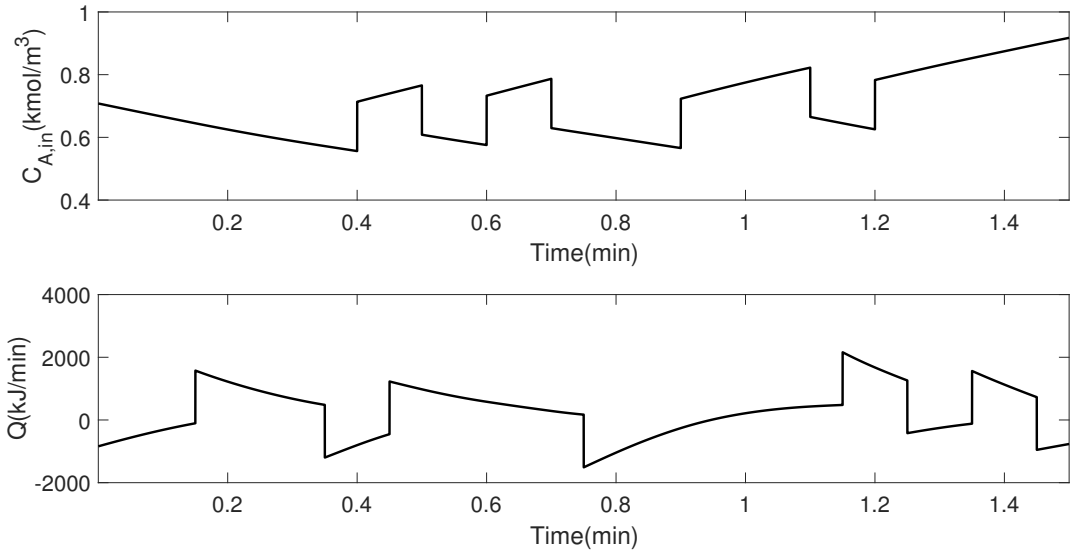

Figure 4.2: Model training data: manipulated inputs under PI controller

For the model validation, initially a steady state Kalman filter (gain calculated by the identification method) is utilized to update state estimate until $t = 0.8min$ and after convergence of the states (gauged via convergence of the outputs), the model and the input trajectory (without the state

Table 4.2: List of controllers parameters for the CSTR reactor

Variable	Value
Δt	$0.2min$
Q_x	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$Q_{x,MPC}$	$10 \times diag([1/C_{A,s}, 1/T_{R,S}])$
$R_{\Delta u,MPC}$	$diag([1/C_{A0,max}, 1/Q_{max}])$
Q_K	$diag([10^3, 10^3])$
R_K	$diag([10^{-3}, 10^{-3}])$
τ_{min}	0
τ_{max}	5
ε_i	$10^{-3} \times x_{i,Sp}$
Δu_{min}	$\begin{bmatrix} -0.1 & -200 \end{bmatrix}$
Δu_{max}	$\begin{bmatrix} 0.1 & 200 \end{bmatrix}$
ϵ^*	1
$V(x)$	$(x - x_{sp})^T (x - x_{sp})$
c_y	$\begin{bmatrix} 10^8 & 0 \end{bmatrix}^T$
c_u	$\begin{bmatrix} 0 & 0.1 \end{bmatrix}^T$
ρ	7.83×10^5

estimator) is used to predict the future output. Figure 4.1 illustrates the results of the model validation, and compares against a model obtained from open-loop step PRBS on the input. As expected, the model identified using closed-loop data predicts better.

Next, closed-loop simulation results for proposed controller and conventional MPC (i.e., MPC without Lyapunov constraint) with horizons 1 and 10 are presented in Figures (4.4-4.7). As can be seen, the LMPC has the best performance in stabilizing the system at the unstable equilibrium point. The MPC with horizon of 1 is not capable of stabilizing the system, and the controller with horizon of 10 reaches set-point later compared to the LMPC. Also the evolution of the subspace states indicates better performance under the proposed LMPC.

4.5 Data-Driven EMPC Design and Illustration

Having illustrated the ability of the LMPC to achieve stabilization, it is next utilized to achieve economical objectives while ensuring stability. The Lyapunov- based EMPC formulation is as follows:

$$\max_{\tilde{u}_k,\ldots,\tilde{u}_{k+P}} \sum_{j=1}^{N_y} c_y^T \tilde{y}_{k+j} - c_u^T \tilde{u}_{k+j}$$

subject to:

$$\begin{aligned} &\tilde{x}_{k+1} = A\tilde{x}_k + B\tilde{u}_k \\ &\tilde{y}_k = C\tilde{x}_k + D\tilde{u}_k \\ &\tilde{u} \in \mathcal{U}, \qquad \Delta\tilde{u} \in \mathcal{U}_\delta, \qquad \tilde{x}(k) = \hat{x}_l \\ &V(\tilde{x}_{k+j}) \leq \rho \text{ for } j = 1, \ldots, P \end{aligned} \tag{4.49}$$

where the value of ρ dictates the neighborhood that the process states are allowed to evolve within. c_y and c_y indicate output and input cost. Other variables have the same definition as Eq. (4.47).

Remark 4.7. *In recent contributions [1, 2] a Lyapunov-Based EMPC is proposed which utilizes data- driven methods to identify an empirical model for the system where the number of empirical model states is equal to the order of the plant model. In contrast, in the present work, the order of the model is selected based on the ability of the model to fit and predict dynamic behavior over a suitable range of operation, in turn allowing for an EMPC design that can reliably operate over a larger region.*

Remark 4.8. *The EMPC formulation in the present manuscript utilizes a linear form of the cost function for the purpose of illustration. The proposed approach is not limited by this particular choice. Any other form of the cost function, including those where the costs could be time dependent, could be readily utilized within the proposed formulation. In such scenarios, the presence of the*

stability constraints provide the safeguards that allow the EMPC to move the process as needed to achieve economical goals.

Remark 4.9. *The use of linear models in the control design opens up the possibility of utilizing MPC formulations [44, 46] that enable stabilization from the entire null controllable region (the region from which stablization is achievable subject to input constraints). The use of the NCR can, in turn, be utilized to maximize the region over which the EMPC can be implemented, thereby maximizing the potential economic benefit. Such an implementation, however, needs to account for potential plant model mismatch owing to the use of the linear model, and remains the subject of future work.*

Next, the proposed Lyapunov-based EMPC (LEMPC) is implemented on the CSTR simulation example and compared to the LMPC implementation. The closed-loop results are presented in Figures (4.8-4.11). Exploiting the flexibility of operation within a neighborhood of the origin, the LEMPC drives the system to a point on the border of that neighborhood, which happens to be the optimal operating point, instead of the nominal operating point. Figure (4.12) shows the comparison of the LEMCP and LMPC. As expected the LEMPC achieves improved economic returns compared to the conventional MPC.

4.6 Conclusions

In this study, a novel data-driven MPC is developed that enables stabilization at nominally unstable equilibrium points. This LMPC is then utilized within an economic MPC formulation to yield a data driven EMPC formulation. The proposed approach is described and compared against a representative MPC and shown to be able to provide improved closed-loop performance.

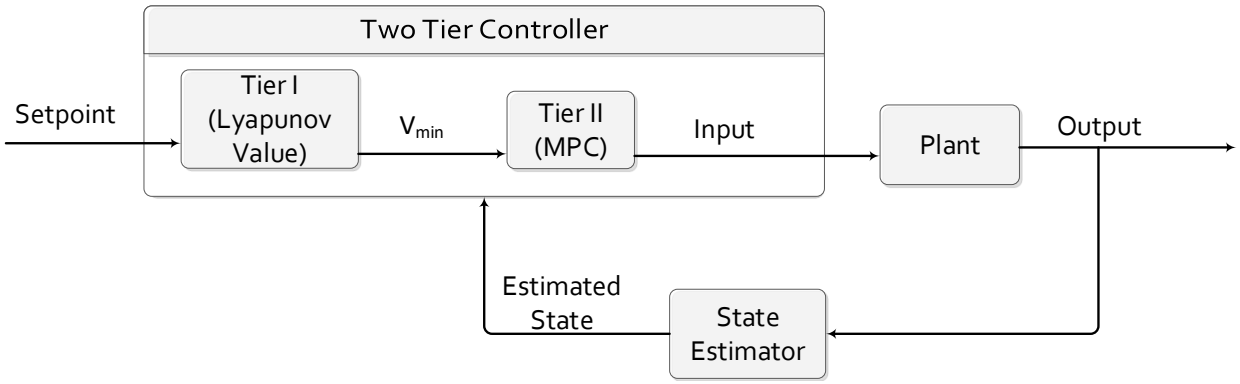

Figure 4.3: Two-tier control strategy

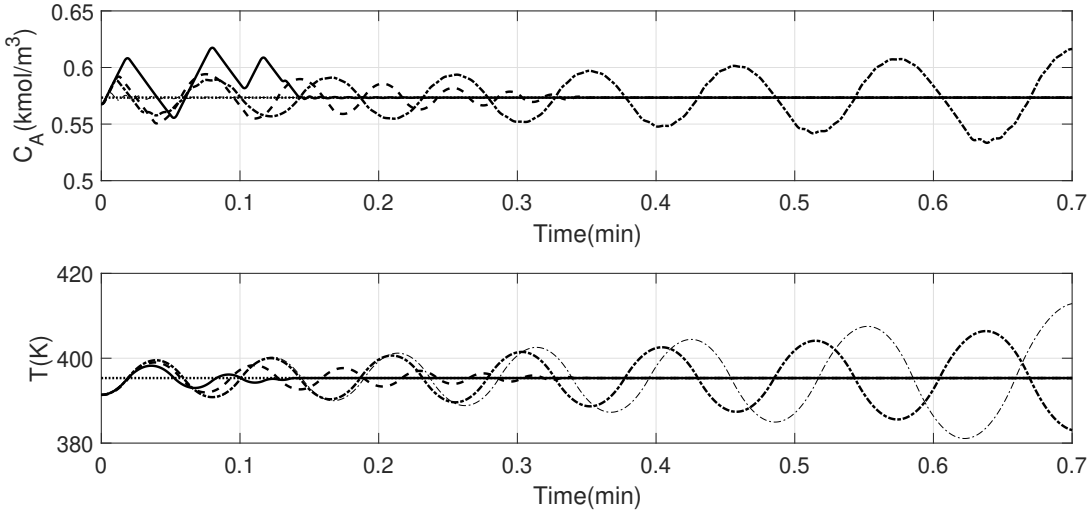

Figure 4.4: Closed-loop profiles of the measured variables obtained from the proposed LMPC (continuous line), MPC with horizon 1 (dash-dotted line), MPC with horizon 10 (dashed line), MPC with horizon 1 and open-loop identification (narrow dash-dotted line) and set-point (dashed line)

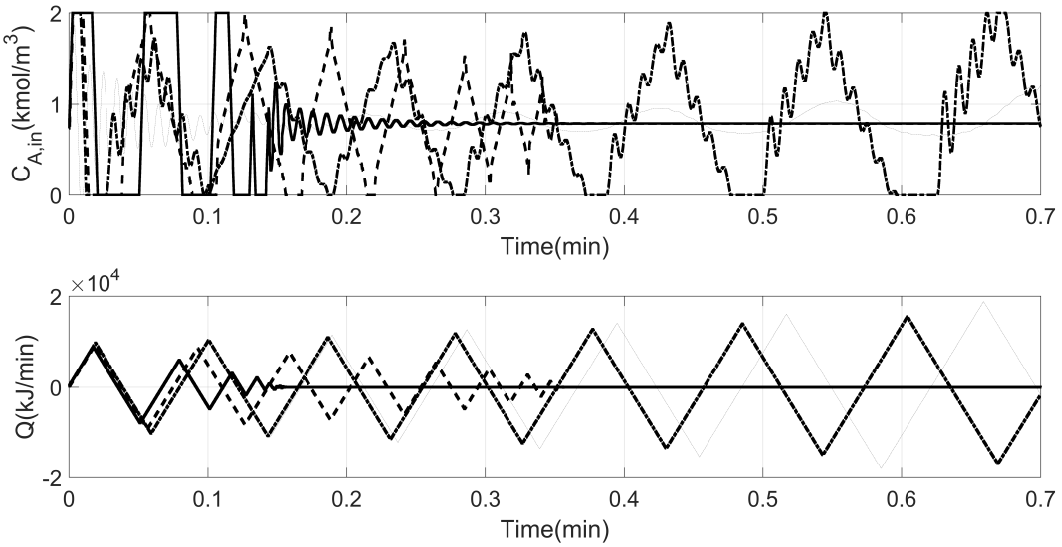

Figure 4.5: Closed-loop profiles of the manipulated variables obtained from the proposed LMPC (continuous line), MPC with horizon 1 (dash-dotted line), MPC with horizon 1 and open-loop identification (narrow dash-dotted line) and MPC with horizon 10 (dashed line)

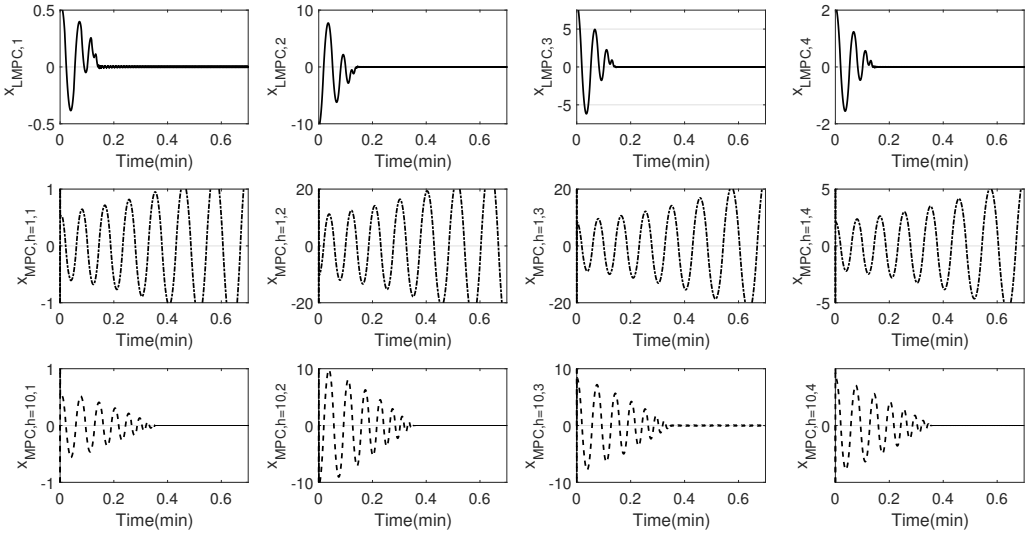

Figure 4.6: Closed-loop profiles of the LTI model states obtained from the proposed LMPC (continuous line), MPC with horizon 1 (dash-dotted line) and MPC with horizon 10 (dashed line)

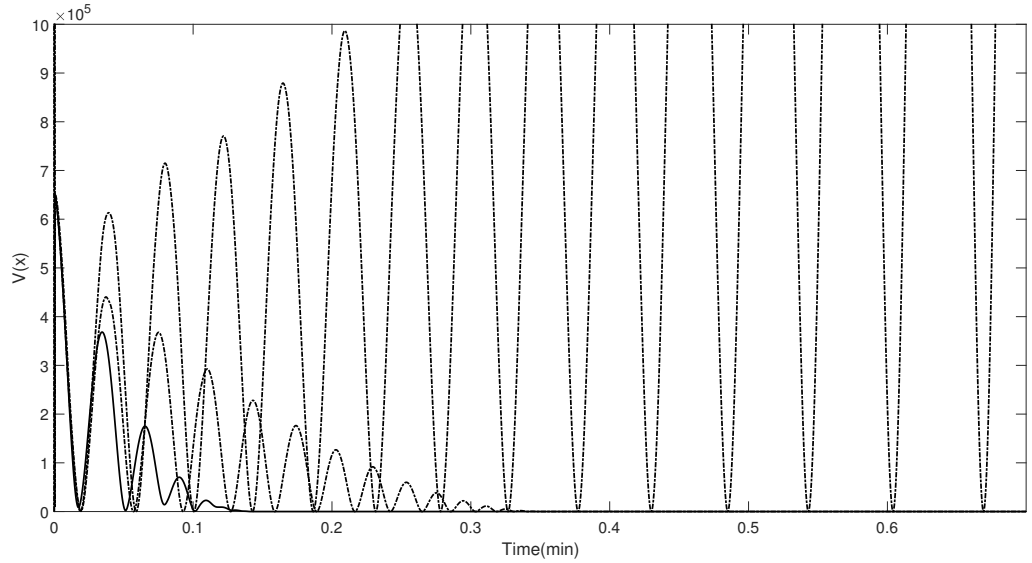

Figure 4.7: Closed-loop Lyapunov function profiles obtained from the proposed LMPC (continuous line), MPC with horizon 1 (dash-dotted line) and MPC with horizon 10 (dashed line)

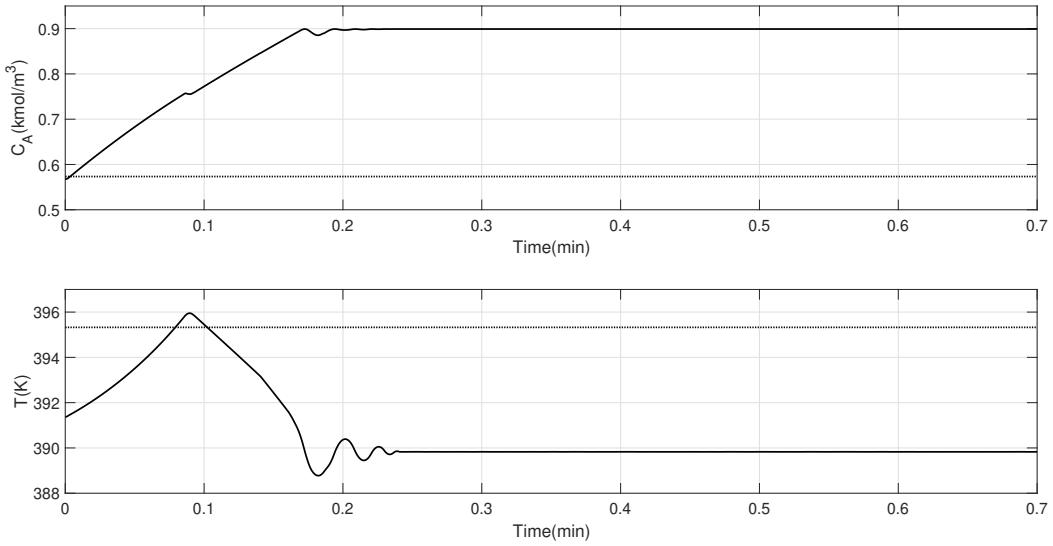

Figure 4.8: Closed-loop profiles of the measured variables obtained from the proposed LEMPC (continuous line) and the nominal equilibrium point (dashed line)

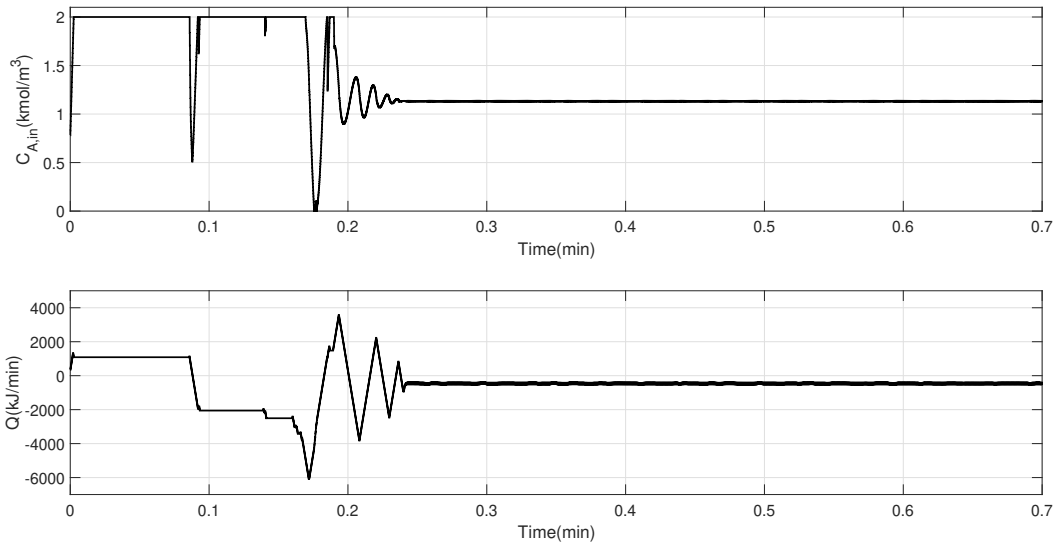

Figure 4.9: Closed-loop profiles of the manipulated variables obtained from the proposed LEMPC (continuous line)

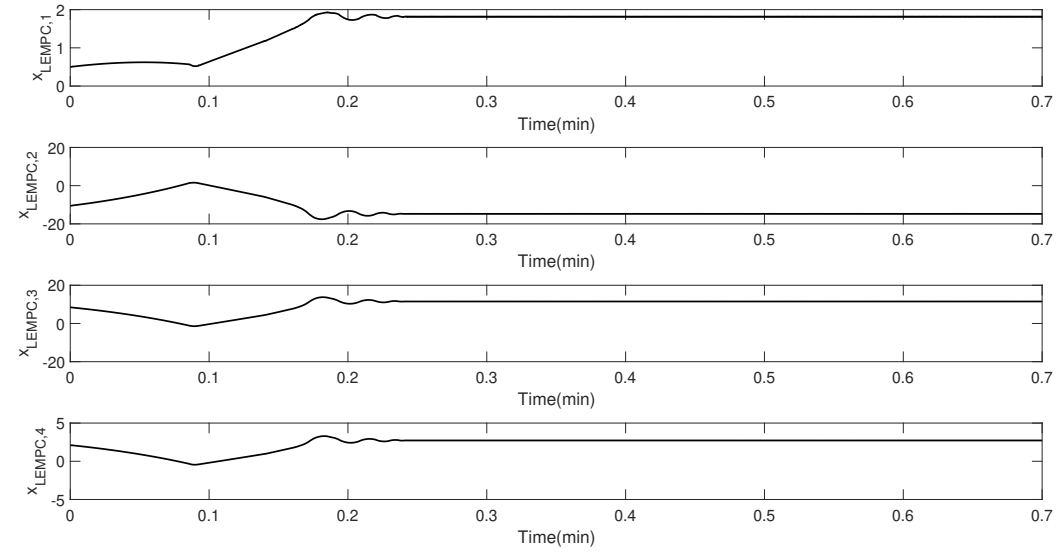

Figure 4.10: Closed-loop profiles of the identified model states obtained from the proposed LEMPC (continuous line)

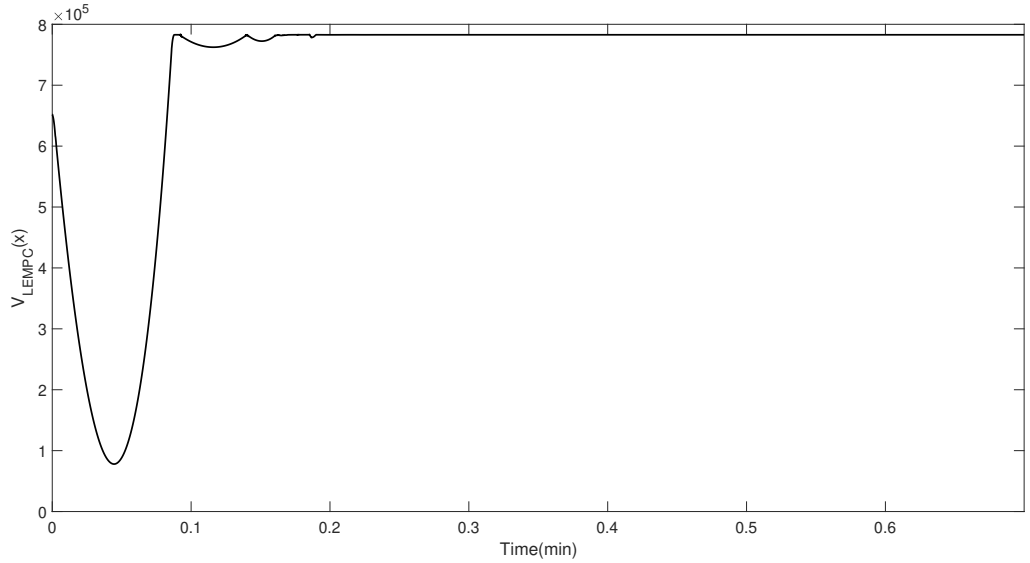

Figure 4.11: Closed-loop Lyapunov function profiles obtained from the proposed LEMPC (continuous line). Note that the LEMPC drives the system to a point within the acceptable neighborhood of the origin.

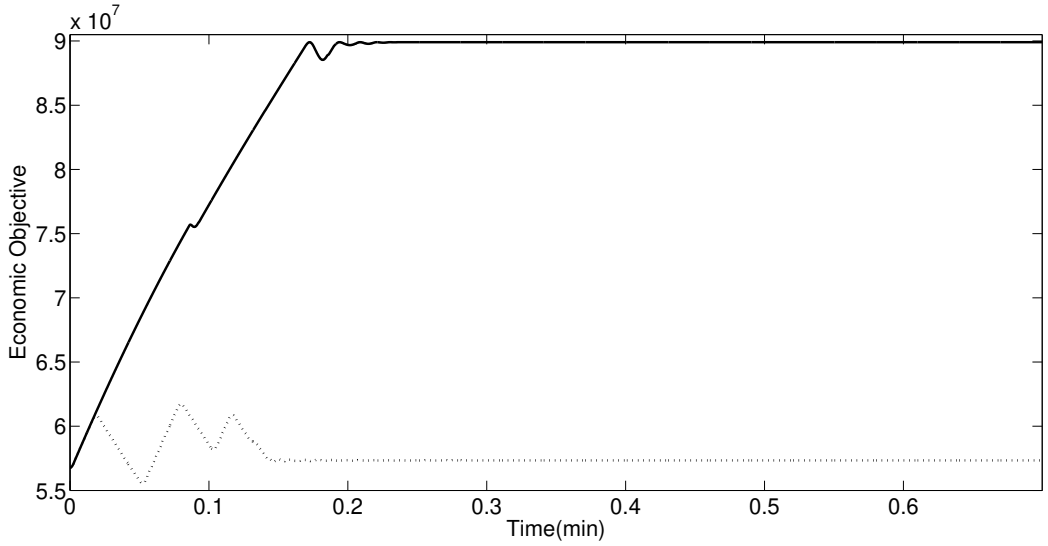

Figure 4.12: A comparison of the economic cost between the LEMPC (continuous line) and LMPC (dotted line).

Chapter 5

PRESCRIBING CLOSED-LOOP BEHAVIOR USING NONLINEAR MODEL PREDICTIVE CONTROL[†]

5.1 Introduction

The operation of chemical plants faces numerous challenges such as inherent nonlinearity, complex variable interactions and process constraints. The presence of these complexities often prevents classical (Proportional Integral Derivative (PID)) controllers from readily achieving closed-loop

[†]The results in this chapter have been published in[34]:

- M. Kheradmandi and P. Mhaskar. Prescribing Closed-Loop Behavior Using Nonlinear Model Predictive Control. *Ind. Eng. Chem. Res.*, 56(51):15083-15093, 2017.
- M. Kheradmandi and P. Mhaskar. Handling Nonlinearity in Model Predictive Control with Explicit Performance Specification. In *Proceedings of the 2018 American Control Conference*, Milwaukee, WI, 2018.

behavior that the control practitioners are looking for (e.g., smooth first order response).

One well established control method that enables incorporating performance considerations more directly in computing control calculations is model predictive control (MPC). In the MPC approach, an open-loop optimal control problem is solved at each sampling instance over a finite time horizon, subject to the dynamic response of the plant and constraints. Contributing to, and benefiting from the industrial application of MPC, several studies have focused on the stability properties of MPC formulations. In one direction, this has led to the development of Lyapunov-based MPC which can explicitly characterize the region from where stability of the closed-loop system is guaranteed in the presence of input constraints [46, 50] and uncertainty [45]. Another direction in exploring the properties of MPC formulations and their ability to handle plant model mismatch has led to the development offset-free MPC designs that focus on the disturbance rejection mechanism by augmenting the state variables with fictitious states that estimate and counteract the uncertainty [13, 57]. In these MPC designs, the focus has primarily been on stabilization, with the objective function being used as a tuning mechanism.

In such cases having desired closed-loop performance is usually done by trial and error. The tuning of the objective function to achieve a good closed-loop performance, however, has remained a non-trivial task.

In this direction, numerous MPC performance/tuning assessment methods are proposed to evaluate the closed-loop performance by comparing the controller with a benchmark [28, 65]. Several excellent contributions have been made to address the challenging problem of controller tuning [20, 21]. In multi-objective MPC (MOPC) [32] the notion of utilizing multi-tier optimization problem is utilized to decouple the tuning issues with the multi-objective optimization problem [74].

The desire to explicitly include economic considerations in the control calculations has fostered

the recent development of economic MPC (EMPC) formulations where the controller determines the set-point internally to satisfy the prescribed economic objective [53], supported by a rigorous analysis that ensures that stability is preserved [4, 25]. In recent contributions, EMPC capabilities for handling constraints, such as limited input rate-of-change [15], while improving economic performance and ensuring closed loop stability have also been addressed [7].

The nature of the predictive controller and existing performance assessment methodologies notwithstanding, controller performance assessment is often conducted by control practitioners in more simpler terms (such as, smoothness of the response or first-order response characteristics). In the direction of computing control laws to deliver specific desire behavior, internal model control (IMC) [19] has been proposed where the controller is designed to achieve a pre-specified closed-loop transfer function for linear, single-input, single-output (SISO) systems. However, it is not always guaranteed that the transfer function of the IMC-based control law will be consistent with the proportional-integral-derivative (PID) control structure. Moreover, the IMC approach does not generalize readily for multi-input, multi-output (MIMO) systems with constraints. In an effort to handle MIMO systems, the funnel control approach [30] is proposed where the time-varying output error feedback controller forces the tracking error to be within a bounded prescribed function. Although the funnel control approach is capable of handling nonlinear MIMO systems, the method does not explicitly consider input constraints.

A related approach is model algorithmic control (MAC), where the desired close-loop trajectory is first order trajectory [56, 64], with the time constant being a tuning parameter. The model algorithmic approach has also been developed for nonlinear systems where the delay free part of the system equations is unstable [55].The MAC approach only allows for prescribing a first order response, and does not explicitly account for the presence of input constraints.

On the other hand, it is common to use the same process for different products in chemical plants, with the different products achieved, say via grade transition in polymerization reactors. This product

transition is usually done by set-point change for plant output[54, 66, 72]. In these instances, a 'desired' process behavior could be specified as one where the transition to the new specification is the fastest, and a resultant optimization problem that minimizes the transition time is formulated and implemented [31].

An MPC framework has recently been proposed that enables specifying desired closed-loop behavior in more general terms for linear MIMO systems subject to input constraints [70], which is then implemented in conjunction with offset-free model predictive control. The developed approach [70] considers systems that are invertible (i.e. the inputs can be explicitly computed). A similar approach was utilized in [73] for linear systems. There does not exist a formulation, however, that allows the ability to explicitly prescribe the nature of the closed-loop behavior and have the formulation determine the best achievable closed-loop behavior for nonlinear systems.

Motivated by the above considerations, in this work we address the problem of control design for nonlinear systems that allows prescribing and determining the best achievable closed-loop behavior of a desired nature. The rest of the manuscript is organized as follows: First, the general mathematical description for the types of nonlinear systems considered in this work, and a representative formulation for nonlinear model predictive control (NMPC) are presented. Then the proposed bi-layer performance specification based nominal MPC scheme for achieving desired trajectories is given. The proposed framework enables specifying a desired nature of the closed-loop behavior and then determining the optimal feasible implementation of such behavior. Rigorous feasibility and stability properties are established for a formulation to achieve the best first order trajectory. Other formulations are also presented that demonstrate how, say a second order trajectory and input rate of constraints can be accommodated. The efficacy of the proposed method is first illustrated through formulations and implementations for a linear system subject to output feedback and a nonlinear continuous stirred-tank reactor (CSTR) with input rate of change constraints and uncertainty and a reactor separator plant. Finally, concluding remarks are presented.

5.2 Preliminaries

In this section, we first describe the class of nonlinear systems considered in this work, followed by a representative existing nonlinear model predictive control formulation.

5.2.1 System Description

In this work, we consider a class of nonlinear systems with input constraints described as follows:

$$\dot{x} = f(x, u) \tag{5.1}$$

$$y = h(x, u) \tag{5.2}$$

where $x \in \mathbb{R}^n$ denotes the vector of state variables, $u \in \mathbb{R}^m$ denotes the vector of constrained control (manipulated) input variables, taking values in a nonempty convex subset $\mathcal{U} \subset \mathbb{R}^m$, where $\mathcal{U} = \left\{u \in \mathbb{R}^m \mid u_{\min} \leq u \leq u_{\max}\right\}$, $u_{\min} \in \mathbb{R}^m$ and $u_{\max} \in \mathbb{R}^m$ denote the lower and upper bounds of the input variables, and $y \in \mathbb{R}^p$ denotes the vector of measured output variables. It is assumed that the functions, $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ are locally Lipschitz in their arguments and that the system is controllable. In keeping with the discrete implementation of MPC, 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.

5.2.2 Nonlinear MPC

In this subsection, we present a representative nonlinear MPC formulation where the control action at time t_k to driven the system to the origin is computed by solving the following optimization problem:

$$\min_{\tilde{u}} \sum_{j=1}^{P} ||\tilde{y}_{k+j} - y_{k+j}^{SP}||_Q^2 + \sum_{j=0}^{N_u-1} ||\tilde{u}_{k+j} - \tilde{u}_{k+j-1}||_R^2 \tag{5.3a}$$

subject to:

$$\dot{\tilde{x}} = f(\tilde{x}, \tilde{u}) \tag{5.3b}$$

$$\tilde{y} = h(\tilde{x}, \tilde{u}) \tag{5.3c}$$

$$\tilde{u} \in \mathcal{U}, \qquad \Delta\tilde{u} \in \mathcal{U}_\delta, \qquad \tilde{x}(k) = \hat{x}_k \tag{5.3d}$$

$$V(\tilde{x}_{k+1}) \leq V(\tilde{x}_k) \; \forall \; V(\tilde{x}_k) > \epsilon^* \tag{5.3e}$$

$$V(\tilde{x}_{k+1}) \leq \epsilon^* \; \forall \; V(\tilde{x}_k) \leq \epsilon^* \tag{5.3f}$$

where $\tilde{x}_{k+j}$, $\tilde{y}_{k+j}$, y_{k+j}^{SP} and $\tilde{u}_{k+j}$ denote predicted state and output, output set-point and calculated manipulated input variables j time steps ahead computed at time step k, and $\hat{x}_k$ is the current estimation of state. The operator $||.||_Q^2$ denotes the weighted Euclidean norm defined for an arbitrary vector x and weighting matrix W as $||x||_W^2 = x^T W x$. Further, $Q > 0$ and $R \geq 0$ denote the positive definite and positive semi-definite weighting matrices for penalizing deviations in the output predictions and for the rate of change of the manipulated inputs, respectively. Moreover, P and N_u denote the prediction and control horizons, respectively, and the input rate of change, given by $\Delta\tilde{u}_{k+j} = \tilde{u}_{k+j} - \tilde{u}_{k+j-1}$, takes values in a nonempty convex subset $\mathcal{U}_\delta \subset \mathbb{R}^m$, where $\mathcal{U}_\delta = \left\{\Delta u \in \mathbb{R}^m \mid \Delta u_{\min} \leq \Delta u \leq \Delta u_{\max}\right\}$. Note finally, that while the system dynamics are described in continuous time, the objective function and constraints are defined in discrete time to be consistent with the discrete implementation of the control action.

Eqs. 5.3e and 5.3f are a representative example of stability constraints (various versions are used in in conventional nominal MPC formulations), and are the Lyapunov-based stability constraint [48, 51], where $V(x_k)$ is a suitable control Lyapunov function, and $\epsilon^* > 0$ is a user-specified parameter. In the presented formulation, $\epsilon^* > 0$ enables practical stabilization to account for the discrete nature of the control implementation.

Note that regardless of the type of stability constraint involved, the objective function is used in conventional MPC formulations to 'tune' the tracking performance. While the tuning parameters can be varied (through an extensive trial and error exercise) to achieve a specific performance, existing MPC formulations do not allow for an explicit specification of the desired closed-loop behavior for general nonlinear systems. In the remainder of the manuscript, the MPC formulation of Eq. 5.3a-5.3f will be referred to as the 'nominal' MPC formulation, and utilized to compare with the proposed formulation.

5.3 MPC Formulation with Performance Specification (PSMPC)

In this section, the MPC formulation that enables specifying and optimizing a desired closed-loop behavior is presented, and is referred to as the performance specification MPC. This desired behavior could be for instance, fastest smooth first-order response, underdamped second-order response and/or a response that is cognizant of the rate of input change constraints. To handle the multi objective nature of the problem the proposed formulation utilizes a bi-layer approach. In the first layer, the best feasible trajectory that satisfies the prescribed desired criteria (subject to appropriate bounds around the desired trajectory) while considering the system dynamics and constraints. Then the second layer of the MPC computes the inputs to track the optimal trajectory satisfying the desired closed-loop behavior as obtained from the first tier (in the form of set-points for the controlled variables along the prediction horizon). Note that the notion of utilizing multi-layers

to decouple the multi objective nature of the optimization problem is not the novel contribution of the present work. Existing results on, for instance, multi-objective MPC [32] have utilized this notion. The novel contribution of the present work is the posing of the control problem where the best achievable desired closed-loop trajectory is computed and implemented.

The two-tiered control structure is schematically presented in Fig. 5.1. We defer the presentation of the specifics regarding Tier 1 to a later section (note that the Tier 1 formulation changes depending on the various considerations pertaining to the desired response that need to be accounted for in the closed-loop). Instead, we first present the detailed formulation for Tier 2 in the following subsection, which implements the specified set-point profile computed by the first tier.

5.3.1 Tier 2: MPC formulation

The second layer of the MPC with explicit performance specification is formulated as follows:

$$\min_{\tilde{u}} \sum_{j=1}^{P} ||\tilde{y}_{k+j} - \bar{y}^{\text{SP}}_{k+j}||^2_{Q_y} \tag{5.4a}$$

subject to:

$$\dot{\tilde{x}} = f(\tilde{x}, \tilde{u}) \tag{5.4b}$$

$$\tilde{y} = h(\tilde{x}, \tilde{u}) \tag{5.4c}$$

$$\tilde{u} \in \mathcal{U}, \qquad \Delta\tilde{u} \in \mathcal{U}_\delta, \qquad \tilde{x}(k) = \hat{x}_k \tag{5.4d}$$

$$V(\tilde{x}_{k+1}) \leq V(\tilde{x}_k) \; \forall \; V(\tilde{x}_k) > \epsilon^* \tag{5.4e}$$

$$V(\tilde{x}_{k+1}) \leq \epsilon^* \; \forall \; V(\tilde{x}_k) \leq \epsilon^* \tag{5.4f}$$

$$|\tilde{y}_{k+j} - \bar{y}^{SP}_{k+j}| \leq \varepsilon, \text{ for } j = 1, ..., P \tag{5.4g}$$

where P denotes the prediction horizon, $\bar{y}_k^{\mathrm{SP}}$ is the desired output trajectory (computed by Tier 1), and, $\tilde{y}_k$ is the predicted output trajectory at time $k\Delta t$ with Δt as the sampling time. $V(\cdot)$ is a control Lyapunov function, $Q_y \in R^{n\times n}$ is a positive definite matrix used to trade-off the relative importance among the controlled variables, and ε is a threshold for maintaining the outputs to be within an admissible neighborhood region of the desired trajectory.

In Fig. 5.2, a schematic presentation of the proposed bi-layer performance specification MPC is presented. The continuous line [−] is the best trajectory that minimizes the key characteristic of the desired closed-loop behavior (say time constant for a first order response), where the process is only required to be within a reasonable bounds of this best trajectory (specified via constraints) denote by the dashed lines [−−]. The corresponding outputs are shown by the dashed square line [−□−]. Thus, there exist a set of feasible input moves that would produce the trajectory denoted by the dashed square lines. The dashed plus line [− + −] is the predicted state trajectory to minimize the difference between the state trajectory and the ideal state trajectory [−]. Thus in the worst case scenario, the dashed plus line [− + −] would simply coincide with the dashed square line [−□−], or could be closer to the continuous line [−], if possible. The lines indicated by [−·] and [··], indicate the Tier 1 and Tier 2 predicted inputs, respectively. Note that the inputs from Tier 1 are feasible for the Tier 2 optimization problem; Tier 2 inputs simply bring the trajectory closer to the ideal trajectory if possible.

Remark 5.1. *The tuning of the response behavior in existing MPC formulations is known to be a nontrivial task. In contrast, the tuning mechanism for the proposed performance specification MPC formulation is significantly less challenging. In particular, the weighting matrix (Q_y) can be readily chosen to scale the variables. Furthermore, since all the performance criteria are already accounted for in Tier 1, the Tier 2 formulation does not include additional penalty terms for the manipulated variables or the input rate of change. Therefore, compared to existing MPC formulations, the trade-off between the penalties is not a hindrance in achieving 'desired' performance in the proposed*

performance specification MPC.

Remark 5.2. *Note that Tiers 1 and 2 are executed in series and at the same time, and the implementation does not require a time scale separation. The overall optimization is split into two tiers simply to enable easy tuning. In particular, the first tier computes the trajectory parameters and input sequence with the objective function only focusing on the 'best achievable' closed-loop response, with constraints in place to allow permissible deviation from the 'best achievable' trajectory. Then the second tier is used to determine the input trajectory that enables closest tracking of the best achievable closed loop response as computed via the constraints of Tier 1 (in other words computing a trajectory that not only respects the allowable deviations from the 'best trajectory' but possibly tracks it better).*

Remark 5.3. *For a class of linear systems that have an equal number of manipulated variables and controlled outputs, recent results[70] incorporated closed-loop performance specification utilizing equality constraints to exploit the fact that there exists a unique steady-state input for each set-point. In contrast the present work considers a general class of nonlinear systems, for which a given closed-loop behavior may simply be unachievable for all the controlled outputs. For instance, it may not be possible for all the closed-loop output responses to be first order, irrespective of the time constant. The resultant optimization problem is also non-convex. Thus, the proposed approach utilizes inequality constraints (Eq. 5.4) to compute and implement the desired closed-loop behavior. The choice of the threshold represents the trade-off between two conflicting objectives, feasibility and efficiency of the first tier. Note further that, for a given value of the tolerance parameter ϵ, infeasibility of the optimization problem can be used by the practitioner to either relax the tolerance, or to ask for a different kind of response (e.g., a second order instead of a first order response).*

Remark 5.4. *Note that there are several key differences between the proposed approach and the MAC approach [56, 64]. The first difference is the capability to handle input constraints. The second key difference is that the proposed formulation allows prescribing other kinds of responses, besides just the first order response. The third key difference is that the proposed approach is configured*

to not just prescribe a specific kind of response, but also prescribe the optimality criteria, for example, fastest first order response. Thus, the proposed formulation not only implements a first order response, but computes and determines the best first order response. Finally, we also show how the proposed approach is able to handle other kinds of constraints (such as rate of change constraints on the manipulated input).

5.3.2 Achieving the Best First Order Trajectory

The key novelty of the proposed approach is that it allows to specify various desired response forms. To illustrate this, in the remainder, we show specific formulations to handle some instances of specific desired behavior. Note that the kinds of possible desired responses is not limited to the ones described next.

We first illustrate the formulation that enables implementing the best first order response, where in this context the best response is one with the smallest time constant. Note that the purpose of the formulation below is to illustrate how such a desired response can be incorporated; if the practitioner dictates other kinds of responses, they can be readily integrated in the approach (see later for other illustrations).

The first tier formulation of the proposed bi-layer control method for a desired first order trajectory

takes the following form:

$$\min_{\tau,\tilde{u}} ||\tau||$$

subject to:

$$\begin{aligned}
&\dot{\tilde{x}} = f(\tilde{x},\tilde{u}) \\
&\tilde{y} = h(\tilde{x},\tilde{u}) \\
&\tilde{u} \in \mathcal{U}, \qquad \Delta\tilde{u} \in \mathcal{U}_\delta, \qquad \tilde{x}(k) = \hat{x}_k \\
&V(\tilde{x}_{k+1}) \leq V(\tilde{x}_k) \; \forall \; V(\tilde{x}_k) > \epsilon^* \\
&V(\tilde{x}_{k+1}) \leq \epsilon^* \; \forall \; V(\tilde{x}_k) \leq \epsilon^* \\
&\bar{y}^{SP}_{i,k+j} = (1 - e^{\frac{-\Delta t}{\tau_i}}) y^{SP}_{i,k+j} + e^{\frac{-\Delta t}{\tau_i}} \bar{y}^{SP}_{i,k+j-1}, \text{ for } i = 1,...,p \\
&|\tilde{y}_{k+j} - \bar{y}^{SP}_{k+j}| \leq \varepsilon, \text{ for } j = 1,...,P, \text{ with } \bar{y}^{SP}_{i,k} = y_{i,k}
\end{aligned} \tag{5.5}$$

where $||\tau||$ is Euclidean norm of vector τ, p is the number of outputs, τ_i is time constant of the ith output. Further, $\bar{y}^{SP}_{i,k}$ is ith output set-point at sample time k based on the desired behavior, $y^{SP}_{i,k}$ is the set-point for ith output, $\hat{x}_k$ is the current estimation of state and y_k is the current output measurement.

In implementation of this optimization based approach, care must be taken to handle any infeasibility related issues. In particular, we recognize that there is no guarantee of initial/successive feasibility of the optimization problem. Initial and successive feasibility of the optimization problem does guarantee closed-loop stability due to the presence of the stability constraint. Furthermore, infeasibility of the optimization problem suggests that from that point in the process, it is simply impossible for all the controlled variables to follow the desired nature of the closed-loop behavior. This in turn can be handled by either requiring the practitioner to request a different kind of closed-loop response, or switching the controller out, and instead implementing a stabilizing controller. In such a scenario, the feasibility of the proposed controller could continue to be checked intermittently,

allowing the controller to be switched on upon being found feasible.

Remark 5.5. *In the formulation, the Euclidean norm is used in the objective function of the Tier 1. Note that if the practitioner chooses, a weighted norm for reflecting the relative importance of the controlled outputs could readily be utilized instead. The use of the weighted norm would not lead to additional tuning parameters, but simply enable the practitioner to specify a 'known'/desired relative importance of the outputs.*

5.3.3 Explicit Tuning Approach for an Underdamped Second Order Specification

As another illustration of the desired behavior, we consider a second order response, described (for each controlled variable) by:

$$\tau_i^2 \frac{d^2 \bar{y}_i^{SP}}{dt^2} + 2\zeta_i \tau_i \frac{d\bar{y}_i^{SP}}{dt} + \bar{y}_i^{SP} = y_i^{SP} \tag{5.6}$$

The solution of this ordinary differential equation with current measured output ($y_{i,0}$), initial derivative $\frac{\partial \bar{y}_i^{SP}}{\partial t}(0) = 0$ and y_i^{SP} as a step function, has the following form:

$$\bar{y}_{i,k}^{SP} = (y_i^{SP} - y_{i,0})(1 - e^{\frac{-\zeta k \Delta t}{\tau_i}}[cos(\frac{\sqrt{1-\zeta_i^2}}{\tau_i} k\Delta t) + \frac{\zeta_i}{\sqrt{1-\zeta_i{}^2}} sin(\frac{\sqrt{1-\zeta_i^2}}{\tau_i} k\Delta t)]) + y_{i,0} \tag{5.7}$$

where $0 \leq \zeta_i < 1$ is the damping factor. Thus the first tier formulated to achieved the best second order trajectory has the following form:

$$\min_{\tau,\zeta,\tilde{u}} ||\tau|| \tag{5.8a}$$

subject to: (5.8b)

$$\dot{\tilde{x}} = f(\tilde{x}, \tilde{u}) \tag{5.8c}$$

$$\tilde{y} = h(\tilde{x}, \tilde{u}) \tag{5.8d}$$

$$\tilde{u} \in \mathcal{U}, \qquad \Delta\tilde{u} \in \mathcal{U}_\delta, \qquad \tilde{x}(k) = \hat{x}_k \tag{5.8e}$$

$$V(\tilde{x}_{k+1}) \leq V(\tilde{x}_k) \; \forall \; V(\tilde{x}_k) > \epsilon^* \tag{5.8f}$$

$$V(\tilde{x}_{k+1}) \leq \epsilon^* \; \forall \; V(\tilde{x}_k) \leq \epsilon^* \tag{5.8g}$$

$$\bar{y}_{i,k}^{SP} = (y_i^{SP} - y_{i,0})(1 - e^{\frac{-\zeta k\Delta t}{\tau_i}}[cos(\frac{\sqrt{1-\zeta_i^2}}{\tau_i}k\Delta t) + \frac{\zeta_i}{\sqrt{1-\zeta_i{}^2}}sin(\frac{\sqrt{1-\zeta_i^2}}{\tau_i}k\Delta t)]) + y_{i,k} \tag{5.8h}$$

$$|\tilde{y}_{k+j} - \bar{y}_{k+j}^{SP}| \leq \varepsilon, \text{ for } j = 1, ..., P \tag{5.8i}$$

with (5.8j)

$$PO_{min} \leq PO \leq PO_{max} \tag{5.8k}$$

$$0 \leq \tau_i, \qquad 0 \leq \zeta_i < 1, \qquad \forall \, i = 1, \ldots, p \tag{5.8l}$$

where PO in the Eq. 5.8a is the percentage overshoot (PO) and PO_{min} and PO_{max} capture the ability to specify allowable ranges. This parameter can be computed as follows:

$$\begin{aligned} PO &= \left[PO_1 \ldots PO_p \right], \\ PO_i &= \frac{-\zeta_i \pi}{\sqrt{1-\zeta_i{}^2}}, \; \forall \, i = 1, \ldots, p \end{aligned} \tag{5.9}$$

Remark 5.6. *The focus in this work is different from economic model predictive controllers or two tier, integrated dynamic optimization and model predictive control problems with economic objectives*[27–30]*. It is not to come up with an economically optimal solution, or to even prescribe a specific kind of performance objective. It is to simply come up with a tool that allows the practitioner to directly prescribe the desired closed-loop behavior. Although economic model predictive controllers compute closed-loop trajectories that are guaranteed to be optimal with respect to the cost function, the closed-loop behavior of EMPC may not always be intuitive or*

admissible to the operator. To address this issue, the propose performance specification MPC can be readily integrated with EMPC. In particular, the Tier 1 could also include a constraint that restricts the feasible region of the optimization to be within an allowable neighborhood of the optimal cost as computed by the EMPC. As such, the trajectory computed by the first layer of the performance specification MPC would simultaneously ensure economic optimality and be prescribed to be a closed-loop behavior that might be more easily acceptable to the process operator.

5.4 Formulations and Simulation Results Handling Specific Instances

In this section we show how various specific issues can be handled within the proposed framework, including implementation under output feedback problem and including rate constraints and uncertainty explicitly in the problem formulation.

5.4.1 Linear System under Output Feedback:

We first consider a the output feedback problem, and for the sake of illustration, consider a linear system of the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{5.10}$$

with the system matrices given in Table 5.1. When implementing the formulation subject to output feedback, the formulation remains the same, except that the state estimates are used in computing the control action. For the purpose of the simulation, we utilize a Luenberger observer to generate state estimates (for nonlinear plants, nonlinear estimators such as moving horizon estimator can be

used [39, 75]). Thus state estimates $\hat{x}_k$ are given as follows:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - C\hat{x}_k) \tag{5.11}$$

where L is the observer gain and is computed using a pole placement method, and y_k is the vector of measured variables. The set point and desired behavior is specified for the measured outputs.

Note that the simulation example presented here is not the same linear system as in [70]. The key difference (beyond the system matrices being different), is that the example in the present manuscript is not a square system so the method proposed in [70] does not remain directly applicable. In particular, for the example in [70], the first order trajectory can be exactly followed and the solution is a unique input that can be computed analytically. In contrast, the input in the present example is non-unique and is computed using an optimization problem to follow the prescribed trajectory subject to an allowable threshold.

The simulation scenario comprises two step changes in the output variable set-points as indicated in Fig. 5.3, 5.4 and 5.5. The tuning parameters of the nominal MPC are varied through a trial and error exercise to achieve a reasonable closed-loop behavior and are reported in Table 5.1. The simulation results show that in both step changes, the proposed controller (with a prescribed desired first order and second order response) provides the prescribed closed-loop behavior as opposed to the nominal MPC (Eq. 5.3a). In particular, note that the nominal MPC results in significant overshoot. In principle, while it may have been possible to find the nominal MPC tuning parameters to give a similar behavior, there is simply no way to specify penalties on overshoot in the nominal MPC formulation. In contrast, the proposed approach enables specifying desired behavior explicitly in the control calculation. Note also that in this case, since the number of inputs is more than the number of outputs, each controller settles at a different steady state manipulated input value. The other key point to recognize is that proposed first and scened order trajectories CPU time required

to solve the two-tired MPC optimization problem were only 75 and 95 percent more than nominal MPC, respectively, and could be further reduced via strategies where the first Tier optimization problem is solved infrequently.

5.4.2 Nonlinear System with Input Rate Constraints and Uncertainty:

We next illustrate the explicit handling of input rate constraints and uncertainty in the proposed approach using the nonlinear CSTR example also used in [70]. To this end, consider a continuous stirred-tank reactor (CSTR) where a first-order, exothermic and irreversible reaction of the form $A \xrightarrow{k} B$ takes place. The mass and energy conservation laws results in the following mathematical model:

$$\begin{aligned} \dot{C}_A &= \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT_R}} C_A \\ \dot{T}_R &= \frac{F}{V}(T_{A0} - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 e^{\frac{-E}{RT_R}} C_A + \frac{Q}{\rho c_p V} \end{aligned} \tag{5.12}$$

The description of the process variables and the values of the system parameters are presented in Table 5.2. The control objective is to track set-point changes using inlet concentration C_{A_0}, and the rate of heat input, Q, while the manipulated inputs are constrained to be within the limits $|C_{A_0}| \leq 1 Kmol/m^3$ and $|Q| \leq 9 \times 10^3 KJ/min$, and the input rate is constrained as $|\Delta C_{A_0}| \leq 0.1 Kmol/m^3$ and $|\Delta Q| \leq 9 \times 200 KJ/min$. We assume that both of the states are measured. Furthermore, there exists plant model mismatch in three of the parameters (5% in c_p, ΔH and 2% in E) and also the inlet temperatures vary around the nominal values by $2K$. To enable a fair comparison, both the the nominal MPC and the proposed two Tier MPC are supplemented with an offset-free mechanism to handle plant-model mismatch. Thus the nonlinear model used in the MPC is augmented with disturbance states, and an extended Kalman filter is utilized to estimate states. The augmented states

result in a model of the form:

$$\begin{aligned}
\dot{C}_A &= \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT_R}} C_A + d_1 \\
\dot{T}_R &= \frac{F}{V}(T_{A0} - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 e^{\frac{-E}{RT_R}} C_A + \frac{Q}{\rho c_p V} + d_2 \\
\dot{d}_1 &= 0 \\
\dot{d}_2 &= 0
\end{aligned} \tag{5.13}$$

Where d_1 and d_2 are disturbance states (see, e.g., [57, 69]).

The extended Kalman filter utilized to estimate the states takes the following form:

$$\begin{aligned}
\dot{\hat{x}}(t) &= f(\hat{x}(t), u(t)) + K(y(t) - h(\hat{x}(t))) \\
\dot{P}(t) &= F(t)P(t) + P(t)F(t)^T - K(t)H(t)P(t) + Q_K \\
K(t) &= P(t)H(t)^T R_K^{-1} \\
F(t) &= \frac{\partial f}{\partial x}|_{\hat{x}(t)} \\
H(t) &= \frac{\partial h}{\partial x}|_{\hat{x}(t)}
\end{aligned} \tag{5.14}$$

Where $\hat{x}(t)$ is the current state estimation, Q_K and R_K are state and measurement covariance matrices. The controller parameters are presented in Table 5.3.

The comparison of simulation results for second order and first order explicit trajectories and nominal MPC are presented in Fig. 5.6 and 5.7. The first and second order explicit trajectories for the performance specification controller are similar, and both the trajectories reach the set-point faster than the nominal MPC without any notable overshoot and with a smooth behavior. The implementation thus demonstrates the key point that the present formulation can be readily adapted to handle various forms of the desired characteristics (such as the additional requirement to handle rate of change of input constraints), and other practical considerations (such as handling

uncertainty). Furthermore, the CPU time required to solve the two-tired MPC optimization problem were only 3 and 13 percent more than nominal MPC, respectively.

Remark 5.7. *One of the existing challenges with nonlinear MPC industrial implementation is the associated computational effort, where the difficulty is further compounded by the tuning effort to achieve a 'desirable' performance. By enabling prescription of the 'desirable' performance in the formulation itself (without increasing the computational complexity significantly), the proposed formulation is expected to alleviate the tuning issues, and thus make it easier to implement NMPC industrially. Note also that the proposed formulation does not comprise two entirely separate optimization problems because the second optimization problem (Tier II) is initialized with a feasible guess (provided by the solution from Tier I), thus limiting the increase in computational complexity.*

5.5 Application to a Reactor-Separator Plant

Finally we implement the proposed MPC formulation on a nonlinear reactor-separator process example with three unit operations. The plant includes two CSTRs and one flash tank separator as shown in Fig. 5.8. The pure reactant A with molar flow rates F_{10} and F_{20} and temperatures T_{10} and T_{20} enter the CSTRs to produce the desired product B, which may react further to form an undesired side-product C. Specifically, reactions of the form $A \xrightarrow{r_1} B \xrightarrow{r_2} C$ take place. The outlet of the first reactor is fed into CSTR 2. The effluent of CSTR 2 is flashed in a heated flash tank. the vapor retrieved from the separator splits into two streams, one is recycled back to reactor 1 at a flow rate F_r, while the other at a flow rate F_p is the plant product. Each vessel is equipped with a jacket to provide/remove heat to/from the vessel. All the three vessels are assumed to have static holdup and

because of the short residence time, there is no reaction taking place in the separator.

$$
\begin{aligned}
\dot{x}_{A1} &= \frac{F_{10}}{V_1}(x_{A10} - x_{A1}) + \frac{F_r}{V_1}(x_{Ar} - x_{A1}) - k_1 e^{\frac{-E_1}{RT_1}} x_{A1} \\
\dot{x}_{B1} &= \frac{F_{10}}{V_1}(x_{B10} - x_{B1}) + \frac{F_r}{V_1}(x_{Br} - x_{B1}) + k_1 e^{\frac{-E_1}{RT_1}} x_{A1} - k_2 e^{\frac{-E_2}{RT_1}} x_{B1} \\
\dot{T}_1 &= \frac{F_{10}}{V_1}(T_{10} - T_1) + \frac{F_r}{V_1}(T_3 - T_1) + \frac{(-\Delta H_1)}{\rho c_p} k_1 e^{\frac{-E_1}{RT_1}} x_{A1} + \frac{(-\Delta H_2)}{\rho c_p} k_2 e^{\frac{-E_2}{RT_2}} x_{B1} + \frac{Q_1}{\rho c_p V_1} \\
\dot{x}_{A2} &= \frac{F_1}{V_2}(x_{A1} - x_{A2}) + \frac{F_{20}}{V_2}(x_{20} - x_{A2}) - k_1 e^{\frac{-E_1}{RT_2}} x_{A2} \\
\dot{x}_{B2} &= \frac{F_1}{V_2}(x_{B1} - x_{B2}) + \frac{F_{20}}{V_2}(x_{B20} - x_{B2}) + k_1 e^{\frac{-E_1}{RT_2}} x_{A2} - k_2 e^{\frac{-E_2}{RT_2}} x_{B2} \\
\dot{T}_2 &= \frac{F_1}{V_2}(T_1 - T_2) + \frac{F_{20}}{V_2}(T_{20} - T_2) + \frac{(-\Delta H_1)}{\rho c_p} k_1 e^{\frac{-E_1}{RT_2}} x_{A2} + \frac{(-\Delta H_2)}{\rho c_p} k_2 e^{\frac{-E_2}{RT_2}} x_{B2} + \frac{Q_2}{\rho c_p V_2} \\
\dot{x}_{A3} &= \frac{F_2}{V_3}(x_{A2} - x_{A3}) + \frac{F_r + F_p}{V_3}(x_{Ar} - x_{A3}) \\
\dot{x}_{B3} &= \frac{F_2}{V_3}(x_{B2} - x_{B3}) + \frac{F_r + F_p}{V_3}(x_{Br} - x_{B3}) \\
\dot{T}_3 &= \frac{F_2}{V_3}(T_2 - T_3) + \frac{Q_3}{\rho c_p V_3}
\end{aligned}
\tag{5.15}
$$

The definitions of the process variables and the parameter descriptions and values are given in Table 5.4, Table 5.5 and Table 5.7. It is also assumed that the relative volatility for each of the components remains constant throughout the operating temperature range of the flash tank separator. The algebraic equations of the composition of the overhead stream relative to the composition of the liquid in the separator is described as follows:

$$
\begin{aligned}
x_{Ar} &= \frac{\alpha_A x_{A3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}} \\
x_{Br} &= \frac{\alpha_B x_{B3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}} \\
x_{Cr} &= \frac{\alpha_C x_{C3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}}
\end{aligned}
\tag{5.16}
$$

The manipulated variables are Q_1, Q_2, Q_3, F_{20}, and the nominal values of the plant are presented in

Table. 5.6. The control objective is to track desired temperature of the three reactors. We assume that full state measurements are available, however, the desired behavior is only prescribed for some of the variables. The simulation results are presented in Fig. 5.9 and 5.10.

To provide a quantifiable comparison of the closed-loop performance in terms of the metric consistent with the prescribed behavior, the closed-loop trajectories of CSTR-Separator simulation example were fitted to a first order trajectory and time constants of these trajectories are reported in Table 5.8. It can be seen that the time constants of the proposed method are either the same or better than the nominal MPC. More importantly, the proposed MPC provides closed-loop trajectories without overshoot.

Note that some of the trajectories under the nominal MPC reach the set point faster compared to the first order or second order trajectories. The key is to recognize, however, that the objective with the control design was not to reach the set point fastest for one of the controlled variables, or all of the controlled variables, but rather was chosen to be the fastest first order behavior for all three controlled variables or the fastest second order response for all three controlled variables. From this standpoint, a faster (but not first order) response in one of the controlled variables with an overshoot in the other controlled variables would not qualify as a better (or to be more precise, close to the prescribed) closed loop behavior. Thus the simulation result demonstrate the significantly improved performance (adherence to the prescribed closed-loop behavior) under the proposed control design, and applicability to larger systems. Furthermore, the CPU time required to solve the two-tired MPC optimization problem were only 60 and 50 percent more than nominal MPC, respectively. All the simulation examples are done in MATLAB, the optimization problems were solved using fmincon function(A local solver), the decision variable is input sequence along the horizon, so the number of decision variables are $n_u \times P$. The ordinary differential equations are solved using ode45 function.

5.6 Conclusion

In this work, a novel MPC based approach is developed that allows specifying desired process behavior, subject to nonlinearity, constraints and uncertainty. The proposed approach is described and compared against traditional nominal MPC and shown to be able to provide desired closed-loop behavior through implementation on three examples that include a non-square linear model subject to output feedback, a CSTR reactor subject to input constraints and uncertainty and a reactor separator process example.

Table 5.1: Parameters for the linear system

Variable	Value
A	$\begin{bmatrix} -6 & -1 & 1 \\ 3 & -5 & 3 \\ 4 & 1 & -2 \end{bmatrix}$
B	$\begin{bmatrix} -2 & -1 & -5 \\ 4 & 2 & 7 \\ -2 & -3 & 3 \end{bmatrix}$
C	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
$x(0)$	$\begin{bmatrix} -1.73 & 3.84 & 3.46 \end{bmatrix}$
u_{min}	$\begin{bmatrix} -100 & -110 & -120 \end{bmatrix}$
u_{max}	$\begin{bmatrix} 150 & 140 & 130 \end{bmatrix}$
PO_{min}	$\begin{bmatrix} 1.00 \times 10^{-2} & 1.00 \times 10^{-2} & 1.00 \times 10^{-2} \end{bmatrix}$
PO_{max}	$\begin{bmatrix} 0.25 & 0.25 & 0.25 \end{bmatrix}$
Δt(s)	5×10^{-2}
Q_y	diag $\{1, 1, 1\}$
τ_{max}(s)	1
ε	$0.1 \times \hat{x}_{Sp,i}$
$Q_{y,MPC}$	diag $\{1, 1, 1\}$
$R_{\Delta u,MPC}$	diag $\{100, 10, 100\}$
P	8
ϵ^*	1
$V(x)$	$(x - x_{sp})^T (x - x_{sp})$
Poles	$\begin{bmatrix} 0.66 & 0.52 & 0.8 \end{bmatrix}$

Table 5.2: Variable and parameter description and values for the CSTR example

Variable	Description	Unit	Value
$C_{A,S}$	Nominal Value of Concentration	$\frac{kmol}{m^3}$	0.573
$T_{R,S}$	Nominal Value of Reactor Temperature	K	395
F	Flow Rate	$\frac{m^3}{min}$	0.2
V	Volume of the Reactor	$\frac{m^3}{min}$	0.2
$C_{A0,S}$	Nominal Inlet Concentration	$\frac{kmol}{m^3}$	0.787
k_0	Pre-Exponential Constant	$-$	72×10^9
E	Activation Energy	$\frac{kJ}{mol}$	8.314×10^4
R	Ideal Gas Constant	$\frac{kJ}{kmol.K}$	8.314
T_{A0}	Inlet Temperature	K	352.6
ΔH	Enthalpy of the Reaction	$\frac{kJ}{kmol}$	4.78×10^4
ρ	Fluid Density	$\frac{kg}{m^3}$	10^3
c_p	Heat Capacity	$\frac{kj}{kg.K}$	0.239

Table 5.3: List of controllers parameters for the CSTR reactor

Variable	Value
Δt	$0.2min$
Q_x	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$Q_{x,MPC}$	$10 \times diag([1/C_{A,s}, 1/T_{R,S}])$
$R_{\Delta u,MPC}$	$diag([1/C_{A0,max}, 1/Q_{max}])$
Q_K	$diag([10^3, 10^3])$
R_K	$diag([10^{-3}, 10^{-3}])$
τ_{min}	0
τ_{max}	5
ε_i	$10^{-3} \times x_{i,Sp}$
Δu_{min}	$\begin{bmatrix} -0.1 & -200 \end{bmatrix}$
Δu_{max}	$\begin{bmatrix} 0.1 & 200 \end{bmatrix}$
ϵ^*	1
$V(x)$	$(x - x_{sp})^T (x - x_{sp})$

Table 5.4: List of CSTR-Separator variables

Parameter	Description
x_{Ij}	Mass fractions of Specie I in Vessel or Stream j; $I = A, B, C$ and $j = 1, 2, 3, r$
T_j	Temperature of Vessel j; $j = 1, 2, 3$
V_j	Volume of Vessel j; $j = 1, 2, 3$
T_{j0}	Temperature of Feed Stream to Reactor j; $j = 1, 2$
F_{j0}	Nominal Feed Flow Rate to Reactor j; $j = 1, 2$
F_j	Outlet Flow Rate from Reactor j; $j = 1, 2$
E_j	Activation Energy for Reaction j; $j = 1, 2$
k_j	Pre-Exponential Constant j; $j = 1, 2$
ΔH_j	Enthalpy of the Reaction j; $j = 1, 2$
α_I	Relative Volatilities for $I = A, B, C$
Q_j	Heat Added to Vessel $j = 1, 2, 3$
c_p	Heat Capacity
R	Gas Constant
ρ	Fluid Density

Table 5.5: CSTR-Separator parameters

Parameter	Value	Parameter	Value
F_{10}	$5.04m^3/h$	Q_3	$1.0 \times 10^6 kJ/h$
F_{20}	$5.04m^3/h$	ΔH_1	$-6.0 \times 10^4 lJ/kmol$
F_r	$50.4m^3/h$	ΔH_2	$-7.0 \times 10^4 lJ/kmol$
V_1	$1.0m^3$	k_1	$2.77 \times 10^3 s^-1$
V_2	$0.5m^3$	k_2	$2.6 \times 10^3 s^-1$
V_3	$1.0m^3$	c_p	$4.2kJ/kmol$
α_A	3.5	R	$8.314kJ/kmol.K$
α_B	1.0	ρ	$1000kg/m^3$
α_C	0.5	x_{A10}	1
T_{10}	$300K$	x_{B10}	0
T_{20}	$300K$	x_{A20}	1
E_1	$5 \times 10^4 kJ/kmol$	x_{B20}	0
E_2	$6 \times 10^4 kJ/kmol$	MW	$242lg/kmol$
Q_1	$2.9 \times 10^6 kJ/h$	Q_2	$0.63 \times 10^6 kJ/h$

Table 5.6: CSTR-Separator controller parameters

Parameter	Value
Δt	$25s$
u_{min}	$\begin{bmatrix} 2.9\times10^5 & 1\times10^5 & 2.9\times10^5 & 0.504 \end{bmatrix}$
u_{max}	$\begin{bmatrix} 5.8\times10^6 & 2\times10^6 & 5.8\times10^6 & 10.8 \end{bmatrix}$
PO_{min}	0
PO_{max}	0.5
$\Delta t(s)$	40
Q_y	$\text{diag}([1/T_{1,s}, 1/T_{2,s}, 1/T_{3,s}])$
τ_{max}(s)	100
τ_{min}(s)	10
ε	$\text{diag}([10^{-3}T_{1,s}, 10^{-3}T_{2,s}, 10^{-3}T_{3,s}])$
$Q_{y,MPC}$	$500\times\text{diag}([1/T_{1,s}, 1/T_{2,s}, 1/T_{3,s}])$
$R_{\Delta u,MPC}$	$\text{diag}([1/Q_{1,s}, 1/Q_{2,s}, 1/Q_{3,s}, 1/F_{20,s}])$
P	15
ϵ^*	1
$V(x)$	$(x-x_{sp})^T(x-x_{sp})$

Table 5.7: CSTR-Separator nominal states and inputs

State	Value	Input	Value
$x_{A1,s}$	0.167	$Q_{1,s}$	$2.9 \times 10^6 kJ/h$
$x_{B1,s}$	0.657	$Q_{2,s}$	$1 \times 10^6 kJ/h$
$T_{1,s}$	$480K$	$Q_{3,s}$	$2.9 \times 10^6 kJ/h$
$x_{A2,s}$	0.191	$F_{20,s}$	$5.04m^3/h$
$x_{B2,s}$	0.637		
$T_{2,s}$	$472K$		
$x_{A3,s}$	0.0571		
$x_{B3,s}$	0.630		
$T_{3,s}$	$475K$		

Table 5.8: Closed-loop time constants (s) for the CSTR-Separator simulation example

Controller	τ_1	τ_2	τ_3
Proposed PSMPC	43.65	13.03	65.36
NMPC	84.64	48.67	94.60

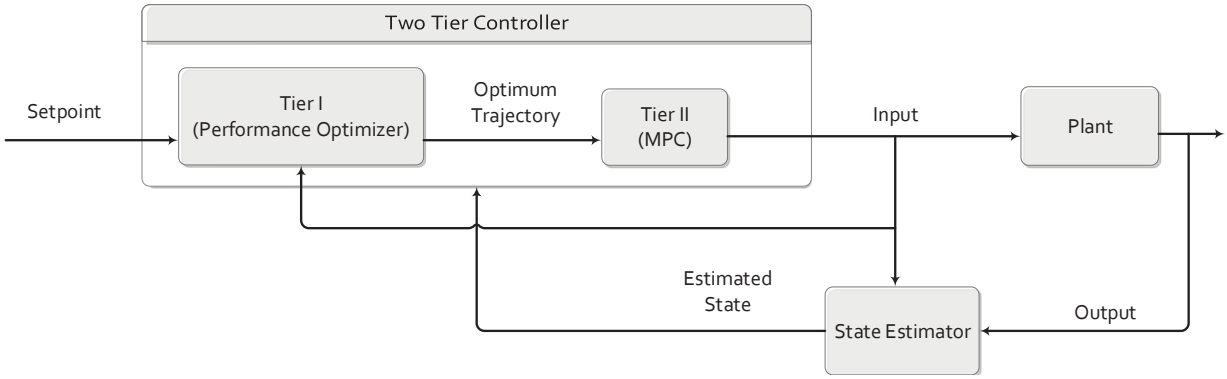

Figure 5.1: Two-tier control strategy

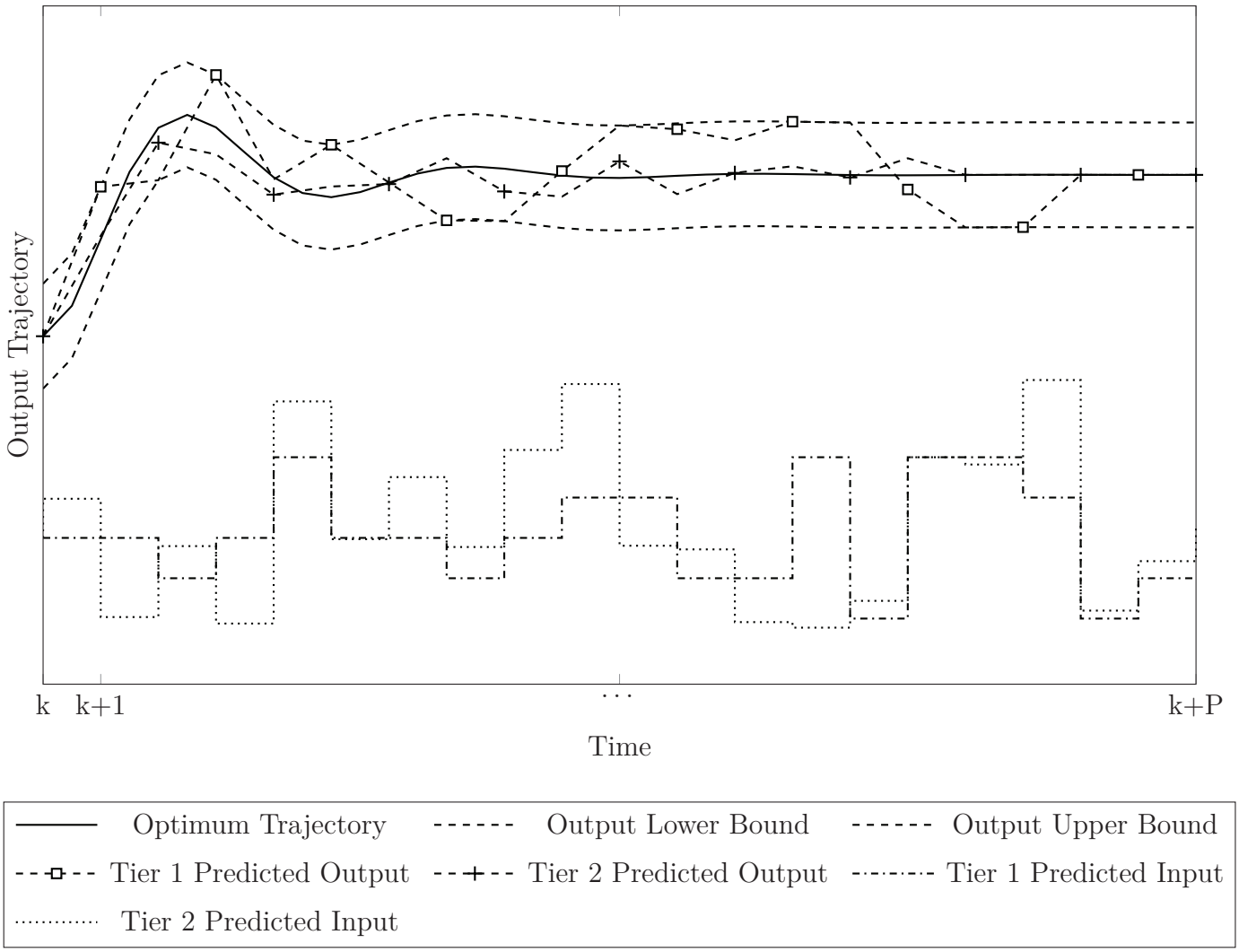

Figure 5.2: Two-tier MPC scheme

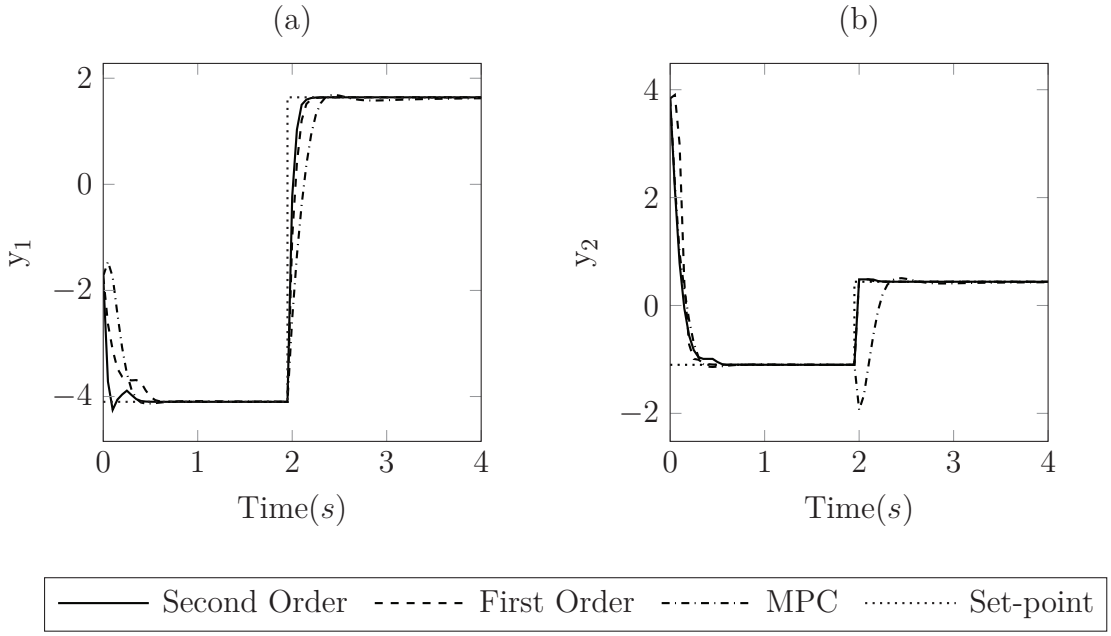

Figure 5.3: Comparison of the proposed MPC approach and Nominal MPC (output variables)

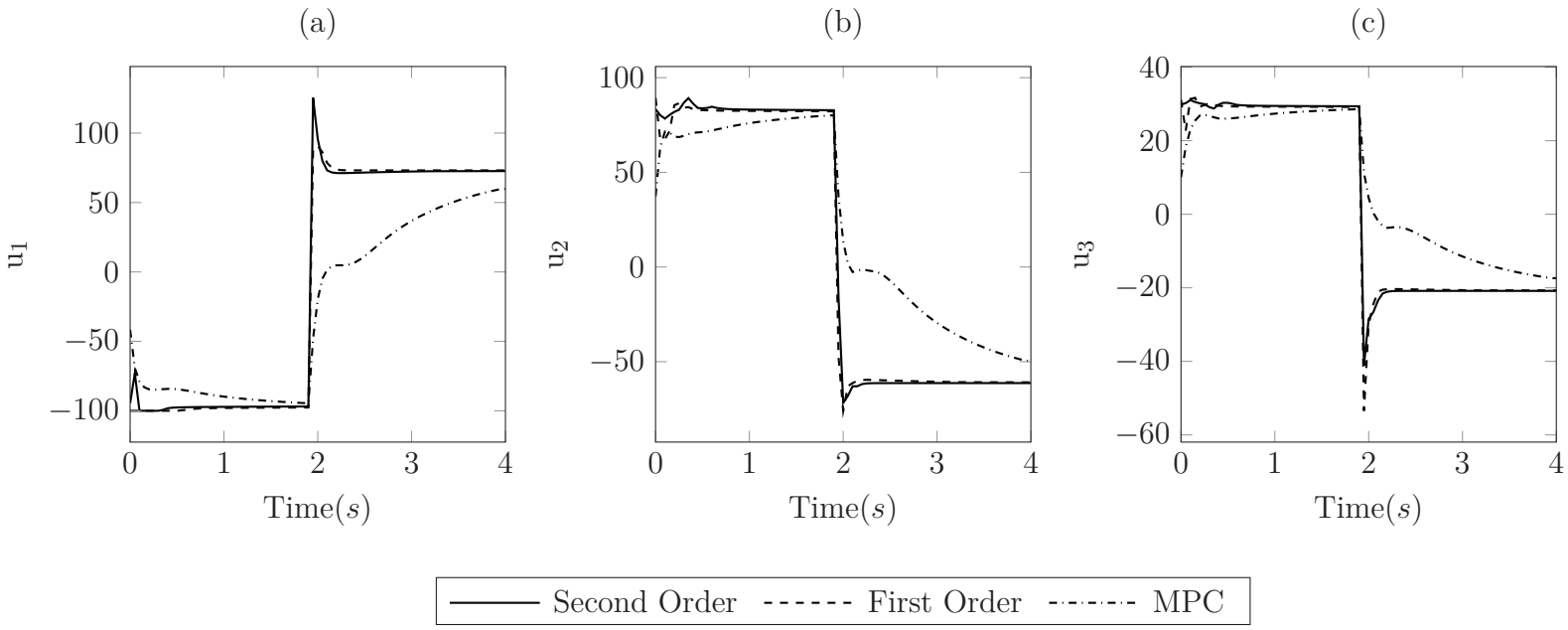

Figure 5.4: Comparison of the proposed MPC approach and Nominal MPC (input variables)

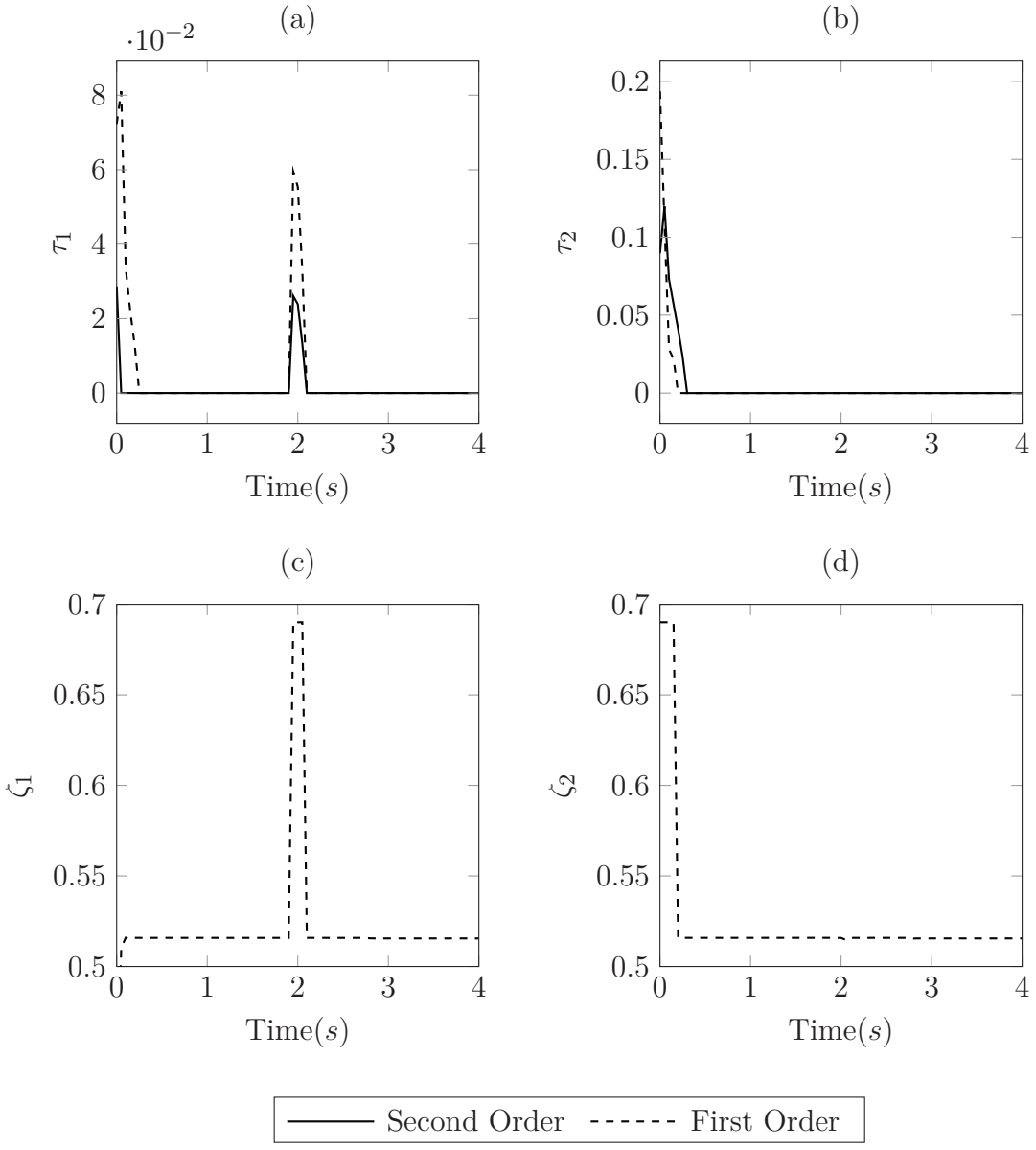

Figure 5.5: Trajectory parameters

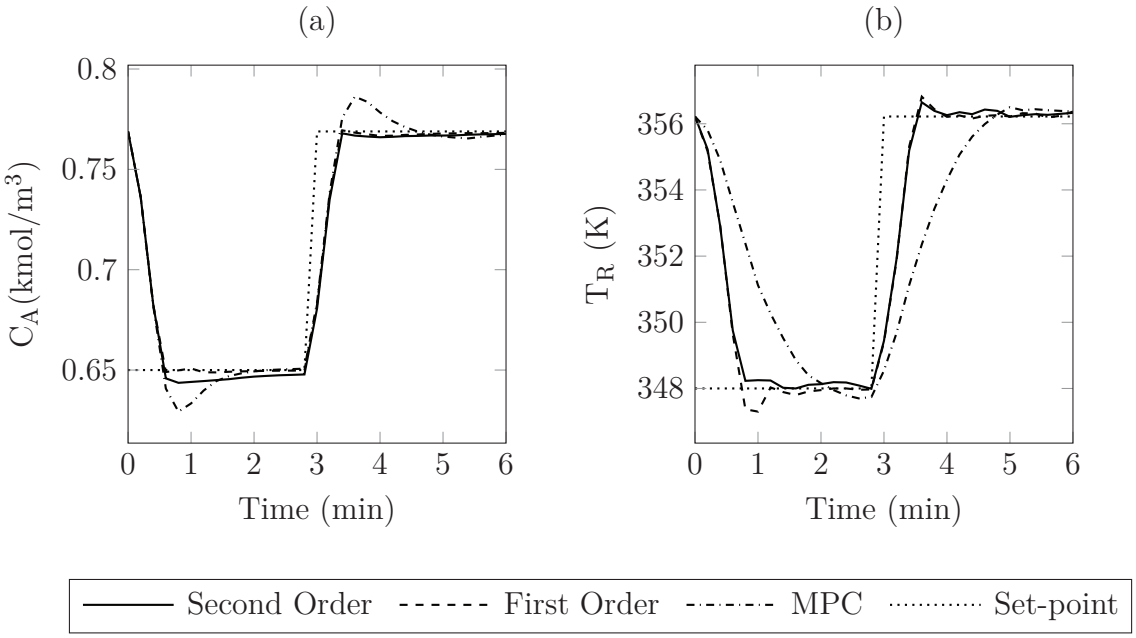

Figure 5.6: Illustrating the proposed approach with input rate constraints and uncertainty on a CSTR example (controlled variables)

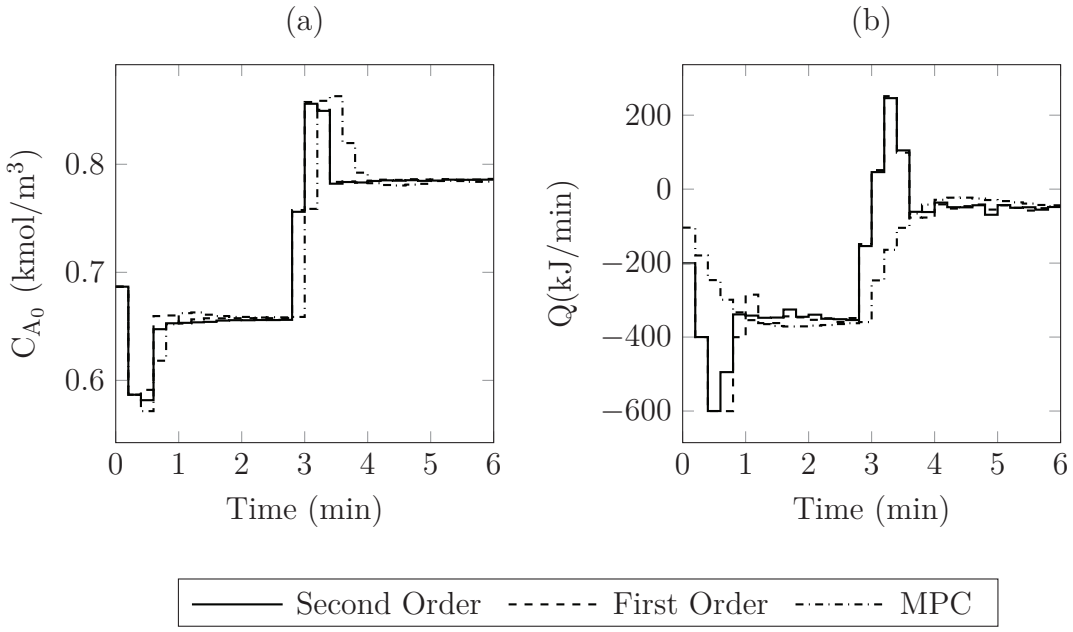

Figure 5.7: Illustrating the proposed approach with input rate constraints and uncertainty on a CSTR example (manipulating variables)

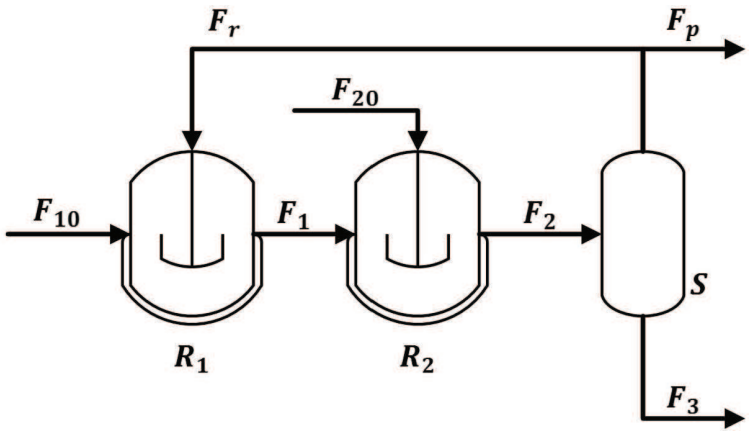

Figure 5.8: A schematic of the reactor separator system

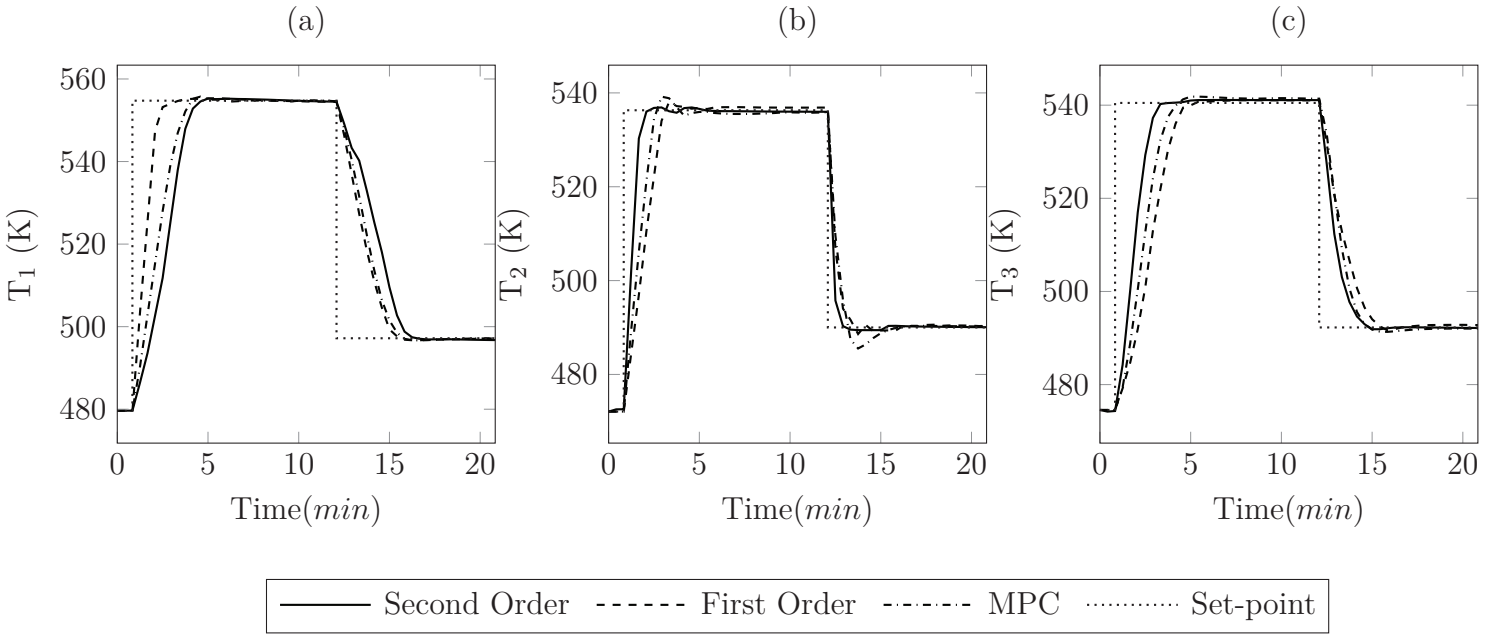

Figure 5.9: Comparison of proposed MPC approach and nominal MPC for CSTR-Separator plant outputs

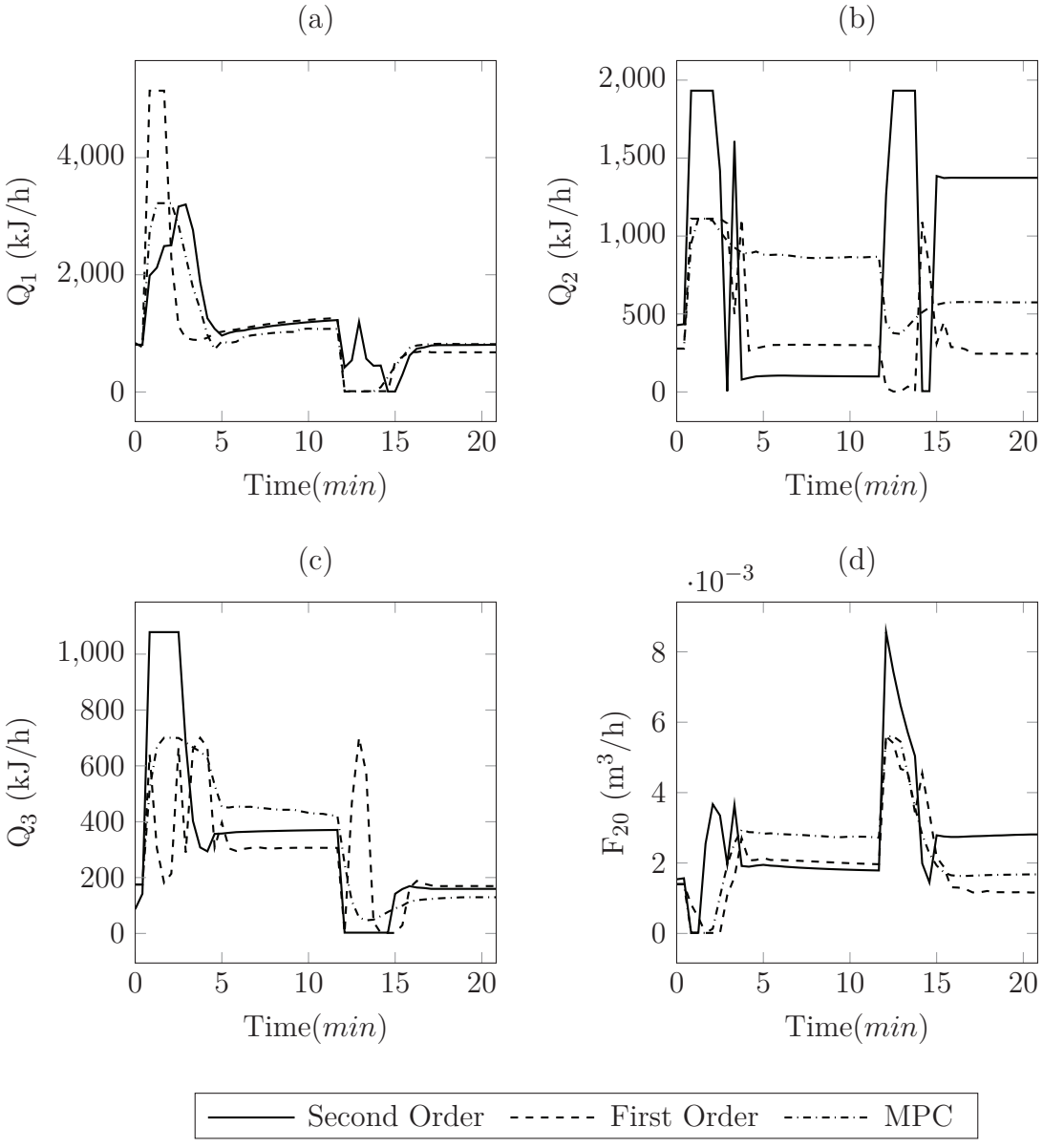

Figure 5.10: Comparison of proposed MPC approach and nominal MPC for CSTR-Separator plant inputs

Chapter 6

CONCLUSIONS AND FUTURE WORK

In this chapter, the subsequent sections summarize the contributions of the research work, followed by suggestions for related future work.

6.1 Conclusions

In an effort to better understand the closed-loop identification and evaluating plant-model mismatch, in Chapter 2, a novel MPC with closed-loop re-identification approach is developed that enables monitoring and updating the model used in the MPC using both past training data and current data. The proposed approach is described and compared against a representative offset-free MPC and shown to be able to provide improved closed-loop behavior through implementation on an example of a polymerization CSTR model subject to measurement noise.

In Chapter 3, a MPC with re-identification approach is developed that enables monitoring and updating the model used in the MPC using both past training data and current data in system identification. The proposed approach is described and compared against traditional nominal MPC

and shown to be able to provide improved closed-loop behavior through implementation on an example of a EAF batch process subject to output feedback and input constraints.

In Chapter 4, a novel data-driven MPC is developed that enables stabilization at nominally unstable equilibrium points. This LMPC is then utilized within an economic MPC formulation to yield a data driven EMPC formulation. The proposed approach is described and compared against a representative MPC and shown to be able to provide improved closed-loop performance.

Finally in Chapter 5, a novel MPC based approach is developed that allows specifying desired process behavior, subject to nonlinearity, constraints and uncertainty. The proposed approach is described and compared against traditional nominal MPC and shown to be able to provide desired closed-loop behavior through implementation on three examples that include a non-square linear model subject to output feedback, a CSTR reactor subject to input constraints and uncertainty and a reactor separator process example.

6.2 Future Work

The contributions and results of this thesis suggest the following topics for future work:

1. Distributed model predictive control with re-identification
2. Utilizing null controllable regions to stabilize data-driven MPC
3. Real-time implementation of MPC with prescribed closed-loop behavior

Below, a summary of the research potential, important developments and main contributions in these future research areas is provided.

The idea of extending MPC with re-identification for distributed systems, can be a future research work. There may be two main challenges in this subject. Firstly, identifying the sub-systems with

poor prediction performance, for re-identifying them and replacing their model. Secondly, for identification purpose recognizing the concept of shared states and subsystems interaction can be a great idea to create a great idea for a strong contribution in the distributed systems identification.

Null controllable region(NCR) is defined as the set of initial conditions from which a constrained system can be stabilized. This region can be utilized for stabilizing control frameworks with mechanistic models but using these methods in data-driven methods is not addressed yet. This work can be a follow up of Chapter 5. Also the effect of re-identification on the NCR can be an interesting challenge to pursue. The challenges can be investigating the effect of model-mismatch handling techniques such as offset-free MPC on NCR.

In another direction, looking into real-time implementation of proposed method in prescribing closed-loop behavior in MPC can be an strong contribution. In the proposed method we need to solve a non-convex optimization problem, presenting associated challenges with the solution.

REFERENCES

[1] Anas Alanqar, Matthew Ellis, and Panagiotis D Christofides. Economic model predictive control of nonlinear process systems using empirical models. *AIChE Journal*, 61(3):816–830, 2015.

[2] Anas Alanqar, Helen Durand, and Panagiotis D Christofides. Fault-tolerant economic model predictive control using error-triggered online model identification. *Industrial & Engineering Chemistry Research*, 56(19):5652–5667, 2017.

[3] LA Alvarez and D Odloak. Optimization and control of a continuous polymerization reactor. *Brazilian Journal of Chemical Engineering*, 29(4):807–820, 2012.

[4] David Angeli, Rishi Amrit, and James B Rawlings. On average performance and stability of economic model predictive control. *IEEE transactions on automatic control*, 57(7):1615–1626, 2012.

[5] Abhijit S Badwe, Ravindra D Gudi, Rohit S Patwardhan, Sirish L Shah, and Sachin C Patwardhan. Detection of model-plant mismatch in mpc applications. *Journal of Process Control*, 19 (8):1305–1313, 2009.

[6] Abhijit S Badwe, Rohit S Patwardhan, Sirish L Shah, Sachin C Patwardhan, and Ravindra D Gudi. Quantifying the impact of model-plant mismatch on controller performance. *Journal of Process Control*, 20(4):408–425, 2010.

[7] Florian A Bayer, Matthias Lorenzen, Matthias A Müller, and Frank Allgöwer. Improving performance in robust economic MPC using stochastic information. *IFAC-PapersOnLine*, 48 (23):410–415, 2015.

[8] G. J. Bekker, I. K. Craig, and P. C. Pistorius. Model predictive control of an electric arc furnace off-gas process. *Contr. Eng. Pract.*, 8(4):445 – 455, 2000.

[9] Dominique Bonvin. Control and optimization of batch processes. *Encyclopedia of Systems and Control*, pages 133–138, 2015.

[10] GA Bustos, A Ferramosca, JL Godoy, and AH González. Application of model predictive control suitable for closed-loop re-identification to a polymerization reactor. *Journal of Process Control*, 44:1–13, 2016.

[11] David Chilin, Jinfeng Liu, James F Davis, and Panagiotis D Christofides. Data-based monitoring and reconfiguration of a distributed model predictive control system. *International Journal of Robust and Nonlinear Control*, 22(1):68–88, 2012.

[12] In S Chin, Kwang S Lee, and Jay H Lee. A technique for integrated quality control, profile control, and constraint handling for batch processes. *Industrial & engineering chemistry research*, 39(3):693–705, 2000.

[13] Buddhadeva Das and Prashant Mhaskar. Lyapunov-based offset-free model predictive control of nonlinear process systems. *The Canadian Journal of Chemical Engineering*, 93(3):471–478, 2015.

[14] David Di Ruscio. A bootstrap subspace identification method: comparing methods for closed loop subspace identification by monte carlo simulations. *Modeling, Identification and Control*, 30(4):203, 2009.

[15] Helen Durand, Matthew Ellis, and Panagiotis D Christofides. Economic model predictive

control designs for input rate-of-change constraint handling and guaranteed economic performance. *Computers & Chemical Engineering*, 92:18–36, 2016.

[16] Jesus Flores-Cerrillo and John F MacGregor. Control of batch product quality by trajectory manipulation using latent variable models. *Journal of Process Control*, 14(5):539–553, 2004.

[17] Jesus Flores-Cerrillo and John F MacGregor. Latent variable mpc for trajectory tracking in batch processes. *Journal of process control*, 15(6):651–663, 2005.

[18] Urban Forssell and Lennart Ljung. Closed-loop identification revisited. *Automatica*, 35(7): 1215–1241, 1999.

[19] Carlos E Garcia and Manfred Morari. Internal model control. a unifying review and some new results. *Industrial & Engineering Chemistry Process Design and Development*, 21(2): 308–323, 1982.

[20] Jorge L Garriga and Masoud Soroush. Model predictive controller tuning via eigenvalue placement. In *2008 American Control Conference*, pages 429–434. IEEE, 2008.

[21] Jorge L Garriga and Masoud Soroush. Model predictive control tuning methods: A review. *Industrial & Engineering Chemistry Research*, 49(8):3505–3515, 2010.

[22] Hasmet Genceli and Michael Nikolaou. New approach to constrained predictive control with simultaneous model identification. *AIChE journal*, 42(10):2857–2868, 1996.

[23] Masoud Golshan, John F MacGregor, Mark-John Bruwer, and Prashant Mhaskar. Latent variable model predictive control (lv-mpc) for trajectory tracking in batch processes. *Journal of Process Control*, 20(4):538–550, 2010.

[24] Iman Hajizadeh, Mudassir Rashid, Kamuran Turksoy, Sediqeh Samadi, Jianyuan Feng, Mert Sevil, Nicole Hobbs, C Lazaro, Zacharie Maloney, Elizabeth Littlejohn, and Ali Cinar. Mul-

tivariable recursive subspace identification with application to artificial pancreas systems. 50: 886–891, 07 2017.

[25] Mohsen Heidarinejad, Jinfeng Liu, and Panagiotis D Christofides. State-estimation-based economic model predictive control of nonlinear systems. *Systems & Control Letters*, 61(9): 926–935, 2012.

[26] Tor Aksel N Heirung, B Erik Ydstie, and Bjarne Foss. Towards dual mpc. *IFAC Proceedings Volumes*, 45(17):502–507, 2012.

[27] Biao Huang and Ramesh Kadali. *Dynamic modeling, predictive control and performance monitoring: a data-driven subspace approach*. Springer, 2008.

[28] Biao Huang and Sirish L Shah. *Performance assessment of control loops: theory and applications*. Springer Science & Business Media, 2012.

[29] Biao Huang, Steven X Ding, and S Joe Qin. Closed-loop subspace identification: an orthogonal projection approach. *Journal of Process Control*, 15(1):53–66, 2005.

[30] Achim Ilchmann, Eugene P Ryan, and Christopher J Sangwin. Tracking with prescribed transient behaviour. *ESAIM: Control, Optimisation and Calculus of Variations*, 7:471–493, 2002.

[31] JV Kadam, W Marquardt, B Srinivasan, and D Bonvin. Optimal grade transition in industrial polymerization processes via nco tracking. *AIChE Journal*, 53(3):627–639, 2007.

[32] Eric C Kerrigan and Jon M Maciejowski. Designing model predictive controllers with prioritised constraints and objectives. In *Computer Aided Control System Design, 2002. Proceedings. 2002 IEEE International Symposium on*, pages 33–38. IEEE, 2002.

[33] Masoud Kheradmandi and Prashant Mhaskar. Model predictive control with closed-loop re-identification. *Computers & Chemical Engineering*, 2017.

[34] Masoud Kheradmandi and Prashant Mhaskar. Prescribing closed-loop behavior using nonlinear model predictive control. *Industrial & Engineering Chemistry Research*, 56(51):15083–15093, 2017.

[35] Masoud Kheradmandi and Prashant Mhaskar. Data driven economic model predictive control. *Mathematics*, 6(4):51, 2018.

[36] Masoud Kheradmandi and Prashant Mhaskar. Model predictive control with closed-loop re-identification. *Computers & Chemical Engineering*, 109:249–260, 2018.

[37] Jay H Lee and Kwang S Lee. Iterative learning control applied to batch processes: An overview. *Control Engineering Practice*, 15(10):1306–1318, 2007.

[38] Weihua Li, Zhengang Han, and Sirish L Shah. Subspace identification for fdi in systems with non-uniformly sampled multirate data. *Automatica*, 42(4):619–627, 2006.

[39] Jinfeng Liu. Moving horizon state estimation for nonlinear systems with bounded uncertainties. *Chemical Engineering Science*, 93:376–386, 2013.

[40] S. Liu and J. Liu. Economic model predictive control with extended horizon. *Automatica*, 73: 180–192, 2016.

[41] Lennart Ljung. System identification. In *Signal analysis and prediction*. Springer, 1998.

[42] JF MacGregor, MJ Bruwer, I Miletic, M Cardin, and Z Liu. Latent variable models and big data in the process industries. *IFAC-PapersOnLine*, 48(8):520–524, 2015.

[43] Urban Maeder, Francesco Borrelli, and Manfred Morari. Linear offset-free model predictive control. *Automatica*, 45(10):2214–2222, 2009.

[44] M. Mahmood and P. Mhaskar. Enhanced stability regions for model predictive control of nonlinear process systems. *AIChE J.*, 54:1487–1498, 2008.

[45] Maaz Mahmood and Prashant Mhaskar. Lyapunov-based model predictive control of stochastic nonlinear systems. *Automatica*, 48(9):2271–2276, 2012.

[46] Maaz Mahmood and Prashant Mhaskar. Constrained control Lyapunov function based model predictive control design. *International Journal of Robust and Nonlinear Control*, 24(2): 374–388, 2014.

[47] Maaz Mahmood, Rahul Gandhi, and Prashant Mhaskar. Safe-parking of nonlinear process systems: Handling uncertainty and unavailability of measurements. *Chemical Engineering Science*, 63:5434–5446, 11 2008.

[48] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[49] David Q Mayne, SV Raković, Rolf Findeisen, and Frank Allgöwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7):1217–1222, 2006.

[50] Prashant Mhaskar, Nael H El-Farra, and Panagiotis D Christofides. Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Transactions on Automatic Control*, 50(11):1670–1680, 2005.

[51] Prashant Mhaskar, Nael H El-Farra, and Panagiotis D Christofides. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Systems & Control Letters*, 55(8):650–659, 2006.

[52] Marc Moonen, Bart De Moor, Lieven Vandenberghe, and Joos Vandewalle. On-and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232, 1989.

[53] Matthias A Müller, David Angeli, and Frank Allgöwer. Economic model predictive control with self-tuning terminal cost. *European Journal of Control*, 19(5):408–416, 2013.

[54] Israel Negrellos-Ortiz, Antonio Flores-Tlacuahuac, and Miguel Angel Gutiérrez-Limón. Product dynamic transitions using a derivative-free optimization trust-region approach. *Industrial & Engineering Chemistry Research*, 55(31):8586–8601, 2016.

[55] Michael Niemiec and Castas Kravaris. Nonlinear model-algorithmic control for multivariable nonminimum-phase processes. *IEE CONTROL ENGINEERING SERIES*, pages 107–130, 2001.

[56] Michael Niemiec and Costas Kravaris. Nonlinear model-algorithmic control: A review and new developments. In *Nonlinear Model Based Process Control*, pages 143–171. Springer, 1998.

[57] Gabriele Pannocchia and James B Rawlings. Disturbance models for offset-free model-predictive control. *AIChE journal*, 49(2):426–437, 2003.

[58] Rohit S Patwardhan and R Bhushan Goapluni. A moving horizon approach to input design for closed loop identification. *Journal of Process Control*, 24(3):188–202, 2014.

[59] Alain Segundo Potts, Rodrigo Alvite Romano, and Claudio Garcia. Improving performance and stability of mpc relevant identification methods. *Control Engineering Practice*, 22:20–33, 2014.

[60] S Joe Qin. An overview of subspace identification. *Computers & chemical engineering*, 30 (10):1502–1513, 2006.

[61] S Joe Qin and Lennart Ljung. Closed-loop subspace identification with innovation estimation. *IFAC Proceedings Volumes*, 36(16):861–866, 2003.

[62] Mudassir M Rashid, Prashant Mhaskar, and Christopher LE Swartz. Handling multi-rate and missing data in variable duration economic model predictive control of batch processes. *AIChE Journal*, 63(7):2705–2718, 2017.

[63] James Blake Rawlings and David Q Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[64] Ramine Rouhani and Raman K Mehra. Model algorithmic control (mac); basic theoretical properties. *Automatica*, 18(4):401–414, 1982.

[65] Sirish L Shah, Rohit Patwardhan, and Biao Huang. Multivariate controller performance analysis: methods, applications and challenges. In *AICHE Symposium Series*, pages 190–207. New York; American Institute of Chemical Engineers; 1998, 2002.

[66] Staffan Skålén, Fredrik Josefsson, and Joakim Ihrström. Nonlinear mpc for grade transitions in an industrial ldpe tubular reactor. *IFAC-PapersOnLine*, 49(7):562–567, 2016.

[67] Peter Van Overschee and Bart De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[68] Peter Van Overschee and BL De Moor. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[69] Matt Wallace, Buddhadeva Das, Prashant Mhaskar, John House, and Tim Salsbury. Offset-free model predictive control of a vapor compression cycle. *Journal of Process Control*, 22(7): 1374–1386, 2012.

[70] Matt Wallace, Steven Spielberg Pon Kumar, and Prashant Mhaskar. Offset-free model predictive control with explicit performance specification. *Industrial & Engineering Chemistry Research*, 55(4):995–1003, 2016.

[71] Jin Wang and S Joe Qin. A new subspace identification approach based on principal component analysis. *Journal of process control*, 12(8):841–855, 2002.

[72] Yang Wang, Hiroya Seki, Satoshi Ohyama, Morimasa Ogawa, Masahiro Ohshima, et al. Opti-

mal grade transition control for polymerization reactors. *Computers & Chemical Engineering*, 24(2-7):1555–1561, 2000.

[73] André Shigueo Yamashita, Paulo Martin Alexandre, Antonio Carlos Zanin, and Darci Odloak. Reference trajectory tuning of model predictive control. *Control Engineering Practice*, 50: 1–11, 2016.

[74] André Shigueo Yamashita, Antonio Carlos Zanin, and Darci Odloak. Tuning the model predictive control of a crude distillation unit. *ISA transactions*, 60:178–190, 2016.

[75] Jing Zhang and Jinfeng Liu. Lyapunov-based MPC with robust moving horizon estimation and its triggered implementation. *AIChE Journal*, 59(11):4273–4286, 2013.

[76] Yu Zhao and S Joe Qin. Subspace identification with non-steady kalman filter parameterization. *Journal of Process Control*, 24(9):1337–1345, 2014.