Economic Model Predictive Control:

Model Approximation and Robustness Analysis

by

Zhiyinan Huang

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

© Zhiyinan Huang, 2023

Abstract

Advanced process control has been considered as a promising tool for addressing various control objectives for complex industrial applications, including ensuring safe operation, reducing operational cost, improving process efficiency, achieving more environmentally friendly practices, etc. Advanced process control strategies often suffer from two major challenges. First, the computational costs for advanced control applications are often very significant. Second, the control performance depends highly on the quality of the embedded model and can be significantly affected by the presence of plant-model-mismatch. Based on the two major challenges, this thesis can be partitioned into two parts. The first part focuses on the investigation of advanced-control-oriented model approximation methods. Specifically, Chapter 2 presents a study on three types of representative model approximation methods applied to economic model predictive control (EMPC), including model reduction based on available first-principle models (e.g., proper orthogonal decomposition), system identification based on input-output data (e.g., subspace identification) that results in an explicitly expressed mathematical model, and neural networks based on input-output data. Two processes that are very different in dynamic nature and complexity were selected as benchmark processes for computational complexity and economic performance comparison, namely an alkylation process and a wastewater treatment plant (WWTP). The strengths and drawbacks of each method are summarized according to the simulation results. In Chapter 3, the observation obtained in Chapter 2 is extended and a two-layer neural network (NN) framework to approximate the dynamics of the agro-hydrological system is proposed. The

model is employed by a EMPC with zone-tracking (ZMPC) that aims to keep the root zone soil moisture in the target zone while minimizing the total amount of irrigation. The performance of the proposed approximation model framework is shown to be better compared to a benchmark long-short-term-memory (LSTM) model for both open-loop and closed-loop applications. Significant computational cost reduction of the ZMPC is achieved with the proposed framework.

The second part of the thesis focuses on the development of generalized ZMPC with guaranteed robustness. In Chapter 4, we propose a generalized robust ZMPC that has guaranteed convergence into the target zone in the presence of bounded disturbance. The proposed approach achieves this by modifying the actual target zone such that the effect of disturbances is rejected. A control invariant set (CIS) inside the modified target zone is used as the terminal set, which ensures the closed-loop stability of the proposed controller. Detailed closed-loop stability analysis is presented. Simulation studies based on a continuous stirred tank reactor (CSTR) are performed to validate the effectiveness of the proposed ZMPC. In Chapter 5, the robust ZMPC proposed in Chapter 4 is improved and applied to an amine-based post-combustion carbon capture plant and is shown to have better performance and enhanced robustness compared to a benchmark ZMPC design.

Preface

The results presented in this thesis are part of the research that is under the supervision of Dr. Jinfeng Liu and Dr. Biao Huang.

Chapter 2 of this thesis is a revised version of Z. Huang, Q . Liu, J. Liu and B. Huang, "A comparative study of model approximation methods applied to economic MPC", The Canadian Journal of Chemical Engineering, 2022;100:1676–1702. I was responsible for conceptualization, control simulation design, and analysis as well as the manuscript composition. Q. Liu was responsible for model reduction conceptualization, simulation, and analysis and contributed to the manuscript composition. J. Liu and B. Huang were the supervisory authors and were involved with the concept formation, manuscript composition and edits.

Chapter 3 of this thesis is a revised version of Z. Huang, J. Liu and B. Huang, "Model predictive control of agro-hydrological systems based on a two-layer neural network modeling framework", International Journal of Adaptive Control and Signal Processing, pp. 1–29, 2023.

Chapter 4 of this thesis is a revised version of Z. Huang, J. Liu and B. Huang, "Generalized Robust MPC with zone-tracking", Chemical Engineering Research and Design, vol. 195, pp. 537–550, Jul. 2023.

Chapter 5 of this thesis is a documentary of the simulation-based application study of the MPC with zone-tracking proposed in Chapter 4, where the process of interest is a Post-Combustion Carbon Capture (PCC) plant.

Acknowledgments

I would like to thank my supervisors, Dr. Jinfeng Liu, and Dr. Biao Huang, for providing me with the opportunity to pursue my studies. They very patiently guided and supported me throughout my study.

I would like to thank all my colleagues from the Process Systems and Control Engineering group and Dr. Huang's research group. They were always there, offering me help and providing me great suggestions. Especially, thank you, Aristarchus Gnanasekar, Benjamin Decardi-Nelson, Bernard Agyeman, Long Wu, Oguzhan Dogru, Song Bo, Siyu Liu, Soumya Sahoo, and Zhuangyu Liu. Also, I would like to thank my very dear friend, Fei Wang for her mental support.

Out of everything, I would like to thank my mother for her unfailing support. This journey would not be possible without her by my side.

I would like to gratefully acknowledge the financial support from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Background 2

1.2.1 An overview of MPC and Its Extensions 2

1.2.2 Advanced Control Oriented Model Approximation 3

1.2.3 Closed-loop Stability and Controller Robustness 4

1.3 Thesis Outline and Contributions 5

2 A comparative study of model approximation methods applied to economic MPC **8**

2.1 Preliminaries 9

2.1.1 System description and problem formulation 9

2.1.2 Model predictive control design 10

2.1.3 Economic model predictive control design 11

2.2 Model Approximation Methods 12

2.2.1 Model reduction based on first-principle model 12

2.2.2 Subspace model identification 15

2.2.3 Neural network modeling 18

2.3 Case Study 1: Benzene Alkylation Process 20

2.3.1 Process overview 20

2.3.2 Control objective and controller setting 21
2.3.3 Data generation and model approximations 23
2.3.4 Results and discussion 26
2.3.4.1 Approximation models 26
2.3.4.2 Conventional tracking model predictive control 27
2.3.4.3 Economic model predictive control 31
2.4 Case Study 2: Wastewater Treatment Plant 34
2.4.1 Process overview 34
2.4.2 Control objective and controller setting 35
2.4.3 Data generation and model approximation 37
2.4.4 Results and discussion 38
2.4.4.1 Approximation models 38
2.4.4.2 Conventional tracking model predictive control 39
2.4.4.3 Economic model predictive control 42
2.5 Summary 44

3 Model predictive control of agro-hydrological systems based on a two-layer neural network modeling framework 47

3.1 Preliminaries 49
3.1.1 Soil Water Dynamics 49
3.1.2 Boundary Conditions 51
3.1.3 Model Discretization 52
3.2 Proposed Two-Layer Neural Network 52
3.2.1 Overview 53
3.2.2 Design and training of first-layer NNs 55
3.2.3 Design and training of second-layer NN 56
3.3 Model Validation 58
3.3.1 Validation of the first-layer sub-models 58

3.3.2 Validation of the proposed two-layer framework 59
3.4 Plant-Model-Mismatch Correction 64
3.5 Controller Design 66
3.6 ZMPC Simulation Results 68
3.6.1 Optimization performance validation 69
3.6.2 Shrinking zone v.s. time-invariant zone 73
3.6.2.1 Tuning $\underline{Y}$ and $\overline{Y}$ 73
3.6.2.2 Tuning μ 77
3.7 Summary 78

4 Robust MPC with zone-tracking **79**
4.1 Preliminaries 80
4.1.1 Notation 80
4.1.2 Description and problem formulation 80
4.1.3 A generalized ZMPC formulation 82
4.2 The Proposed Robust ZMPC 84
4.3 Stability Analysis 86
4.4 An Algorithm for Estimating the Modified Target Set $\tilde{\mathbb{X}}_t$ 92
4.5 Consideration of Economic Objectives 96
4.6 Simulations 97
4.6.1 Process description 98
4.6.2 Controller setup and implementation 99
4.6.3 Control performance validation 100
4.6.4 The significance of modifying the target set 102
4.6.5 The significance of the terminal constraint 103
4.6.6 The impact of γ 104
4.7 Summary 108

5 Robust ZEMPC Application to an amine-based post-combustion CO_2 capture process **109**

5.1 Preliminaries 111

5.1.1 Process Overview 111

5.1.2 Model Description and Control Objective 112

5.2 Robust Economic Model Predictive Control with zone-tracking 114

5.3 Determining the Control Invariant Terminal Set 116

5.3.1 Approximation of CIS in the state space 116

5.3.2 Projecting $\mathbb{X}_I$ into the output space 121

5.4 Simulations 123

5.4.1 Controller setup and implementation 123

5.4.2 Result and discussion 125

5.5 Summary 127

6 Conclusions and Future work **129**

6.1 Conclusions 129

6.2 Future Work 131

List of Tables

2.1 Open-loop prediction performance of approximation models - Alkylation . . 28

2.2 MPC results of the original model and approximation models - Alkylation . 29

2.3 EMPC results of the original model and approximation models - Alkylation . 31

2.4 Open-loop response of the reduced model and the original model - WWTP . 39

2.5 MPC results of the reduced model and the original model - WWTP 41

2.6 EMPC results of the reduced model and the original model - WWTP 44

3.1 Soil parameters of sandy loamy soil 51

3.2 Operating ranges of the sub-models 54

3.3 NRMSE of multi-step-ahead-predictions of the sub-models 58

3.4 NRMSE of multi-step-ahead-predictions of the NN models 63

3.5 Performance of the mismatch correction approach in terms of average absolute error 66

3.6 ZMPC simulation results in the presence of 2% process noise. 71

3.7 ZMPC simulation results in the presence of process noise (P.N.) and measurement noise (M.N.). 71

3.8 ZMPC simulation results with different $\underline{Y}$ and $\overline{Y}$ 73

3.9 ZMPC simulation results with different μ 77

4.1 The number of violations and the average violation value after the first convergence to the actual target set with different γ. 106

5.1 The total zone-tracking cost and economic cost of the proposed robust ZEMPC v.s. the nominal ZEMPC 127

List of Figures

2.1 A schematic of the alkylation of benzene process 20

2.2 Step response of the alkylation process output. 22

2.3 Multi-level input and the corresponding output response. 24

2.4 Open-loop output responses of the alkylation first-principle model and approximation models. 27

2.5 The optimal product concentration trajectory under different model-approximation-based conventional MPC frameworks. 29

2.6 Manipulated input trajectory under different model-approximation-based conventional MPC frameworks - the heat supply going into CSTR-4 30

2.7 The optimal product concentration trajectory under different model-approximation-based EMPC frameworks. 32

2.8 Manipulated input trajectory under different model-approximation-based EMPC frameworks - the heat supply going into CSTR-4 32

2.9 A schematic of the wastewater treatment plant. 34

2.10 Step response of a WWTP process output. 36

2.11 Multi-level input and the corresponding output response. 38

2.12 Open-loop output responses of the WWTP first-principle model and approximation models. 40

2.13 Optimal output trajectory of the WWTP under different model-approximation-based conventional MPC frameworks. 42

2.14 Optimal control policy of the WWTP under different model-approximation-based conventional MPC frameworks. 43
2.15 Effluent quality (EQ) index trajectory under different model-approximation-based EMPC frameworks. 45
2.16 Optimal control trajectories under different model-approximation-based EMPC frameworks 46

3.1 A schematic diagram of the considered agro-hydrological system. 50
3.2 A schematic diagram of the proposed two-layer NN framework. 53
3.3 A segment of the training dataset of M_2 55
3.4 A segment of the training dataset of M_A 57
3.5 One-step-ahead prediction (solid), ten-step-ahead prediction (dash-dotted), and twenty-step-ahead prediction (dashed) of M_1 compared to the actual trajectory (dark solid). 60
3.6 Multi-step-ahead prediction performance of M_2. 60
3.7 Multi-step-ahead prediction performance of M_3. 61
3.8 Multi-step-ahead prediction performance of an single LSTM. 61
3.9 Multi-step-ahead prediction performance of the two-layer NN framework. . . 62
3.10 Optimal control strategy and the corresponding output response of ZMPC in the presence of 2% process noise. Discretized Richards equation (star marker), the single LSTM (dot marker), the two-layer NN framework (triangle marker). 70
3.11 Optimal control strategy and the corresponding output response of ZMPC in the presence of 5% process noise and 5% measurement noise. 70
3.12 Shrinking zones with different $\underline{Y}$ and $\overline{Y}$. 74
3.13 The precipitation data (top), optimal control strategies (middle), and corresponding output response (bottom) of ZMPC with different $\underline{Y}$ and $\overline{Y}$. 75
3.14 Shrinking zones with different μ. 76

3.15 The optimal control strategies (top), and corresponding output response (bottom) of ZMPC with different μ. 76

4.1 Potential nominal state trajectory (dotted line) and actual state trajectory (dashed line) of the closed-loop system under the proposed ZMPC. 85
4.2 A sample diagram of $\mathbb{P}(\mathbb{S})$. 91
4.3 Proposed ZMPC without economic objectives. 100
4.4 Proposed ZMPC v.s. the nominal ZMPC with economic objective, $x_0 = [0.12, 355]$. 101
4.5 Proposed ZMPC v.s. the nominal ZMPC with economic objective, $x_0 = [0.9, 345]$. 102
4.6 ZMPC with economic objective that tracks the original target set ($\mathbb{X}_t$) with modified terminal set ($\mathbb{X}_f = \mathbb{X}_t^M$). 103
4.7 ZMPC without the terminal constraint v.s. the proposed ZMPC with economic objective. 104
4.8 The modified target sets and the corresponding maximum CISs under different γ. 105
4.9 The zone-tracking performance and economic performance under different γ. 106

5.1 A schematic diagram of an amine-based post-combustion carbon capture plant (PCC). 112
5.2 An illustrative diagram showing three potential iterations Algorithm 4 120
5.3 A schematic diagram of the relationship between $\mathbb{Y}_f$ and $\mathbb{X}_I$ 121
5.4 The actual target set, modified target set, and the terminal set of the proposed ZEMPC 124
5.5 The system output trajectories obtained based on the proposed robust ZEMPC v.s. the nominal ZEMPC 125

5.6 The zone-tracking cost over the simulation time obtained based on the proposed robust ZEMPC v.s. the nominal ZEMPC 126
5.7 The economic cost over the simulation time obtained based on the proposed robust ZEMPC v.s. the nominal ZEMPC 126

Chapter 1

Introduction

1.1 Motivation

Due to the development in computational hardware, nonlinear model predictive control (MPC) is considered to be a promising advanced control strategy and has attracted high research interests over the past decades. Inspired by the linear quadratic regulator (LQR), MPC has the ability to solve nonlinear optimization problems while considering constraints simultaneously. However unlike LQR problems that can be solved explicitly for the infinite horizon, the same cannot be achieved for MPC due to the presence of constraints. Instead, MPC is solved in a receding horizon manner [1, 2], where the optimization problem is solved for a finite control horizon at each sampling instant with only the very first control policy applied to the system.

Various extensions of MPC are developed based on different objectives, for example, economic MPC, MPC with zone-tracking, etc. More details regarding MPC extensions are introduced in Section 1.2. Overall, MPC and its extensions all share some bottleneck challenges when it comes to real-world applications. One of the shared challenges is the high computational cost of solving complex nonlinear optimization problems online. The computational burden of MPC rises significantly when the complexity of the embedded model

rises. Thus reducing the model complexity is essential for successful MPC applications and is worth investigating. In this thesis, we investigated the performance of various model approximation and reduction techniques in advanced process control applications. Furthermore, we developed a case-specific model approximation approach for an agro-hydrological system and validated its performance for MPC application.

The performance of MPC and its extensions highly depends on the accuracy of the process model employed inside the controller. However, plant-model-mismatch is often unavoidable in applications. It is thus vital to take this into consideration for controller design and ensure reliable control performance in the presence of plant-model-mismatch, which is also referred to as the robustness of the controller. This also aligns with the root motivations for introducing automated control into processes: ensure operational safety. The investigation of closed-loop stability and robustness of MPC and its extensions has been continuously a vital scope of research. In this thesis, a robust MPC with zone-tracking is proposed with mathematically proven robustness. Furthermore, the performance of this controller is validated on a Post-Combustion Carbon Capture (PCC) process, which is a complex high-dimensional nonlinear system.

1.2 Background

1.2.1 An overview of MPC and Its Extensions

The most basic form of MPC is the conventional MPC, which tracks a referencing trajectory. The referencing trajectory is oftentimes an optimal steady-state operating point [2]. This approach, however, may limit the flexibility of the controller and sacrifice the control performance during transient operations.

One popular extension of the conventional MPC is economic MPC (EMPC), which considers a general economic objective directly in dynamic optimizations [3]. EMPC is recognized as a promising extension of the decision-making framework for the next generation

smart manufacturing [4]. Improved transient economic performance has been observed in EMPC compared to MPC [5, 6].

Another approach that helps to improve the flexibility of MPC/EMPC is to add zone tracking (ZMPC). ZMPC relaxes the set-point-tracking objective to a zone-tracking one, which aims to drive the system into a bounded set [7, 8]. More degrees of freedom are provided by ZMPC, which are beneficial for handling multiple control objectives (e.g. tracking and economics) simultaneously. Furthermore, ZMPC is potentially more robust in the presence of noise or disturbance. Zone-tracking is also a natural objective that arises in many real-world problems. For example, in agricultural practice, maintaining the soil moisture in a certain range is often sufficient and more practical rather than attempting to maintain it at a particular level. Various applications of ZMPC have been reported in the literature, for example, treatment of diabetes in [9, 10], control of heat system inside a building in [11], control of coal-fired boiler-turbine generating systems in [12], and control of irrigation systems in [13] and [14].

1.2.2 Advanced Control Oriented Model Approximation

Solving advanced process control optimization problems often leads to significant computational load, especially in large-scale systems [4, 15]. Model approximation has been considered to be an effective way to reduce the computational burden and improve the applicability of EMPC for large-scale processes [16]. It aims to approximate the original system with an appropriately simplified model that captures the main dynamics of the original system. The popular methods for model approximation can be broadly classified into three categories: model reduction based on first-principle models (e.g., proper orthogonal decomposition) [17], traditional data-based system identification (e.g., subspace identification) [18, 19, 20, 21], and machine-learning-based model identification [22].

Model reduction techniques require the existence of a (typically first-principle) model of the considered system [23, 24]. Many model reduction methods in this category belong to

projection-based methods, in which the original high-order system is projected onto a lower-order reduced space [24]. Among them, the proper orthogonal decomposition (POD) is one of the most widely used methods, which is for example applied in [17] together with the discrete empirical interpolation method to solve large-scale model predictive control (MPC) problems for continuous chemical processes.

In the cases where only the input-output dataset is available, system identification techniques are frequently used to establish the mathematical model of a dynamical system [25, 26], where relatively simple models can be generated without knowledge of the underlying physical laws of the system. There are mainly three types of system identification methods, namely the prediction error method [27], maximum likelihood method [28], and subspace identification method [18]. Different from the first two methods that require pre-selected model structure and solving optimization problems, the subspace identification method can directly extract the state-space model from the input-output data without using optimization techniques [18, 21].

Different from system identification methods, data-driven machine-learning-based model identification typically leads to neural network (NN) models that do not have explicit mathematical expressions [22]. With the ability to acquire knowledge from the input information and remember the learning results, neural networks can approximate any continuous function according to the universal approximation theorem [29]. Much attention has been attracted to neural-network-based system modeling [30, 31] and control [32] recently. However, for various applications, neural networks often suffer from their black-box nature. Thus explicitly expressed NNs were proposed and applied in some areas [33].

1.2.3 Closed-loop Stability and Controller Robustness

Closed-loop stability of the system is always an essential aspect to be considered in controller design. Unstable controllers can lead to drastic safety risks in process operation. The closed-loop stability theories of MPC and EMPC are well-developed and widely used in literature

based on the Lyapunov stability theory [34, 35, 36, 37, 38]. However, most of these stability theories rely on the assumption of the existence of an optimal operating point. On the other hand, ZMPC is often considered to be a relaxation to conventional MPC but with not many investigations done in terms of the stability property. The stability of ZMPC with a secondary economic objective is investigated in [39]. An extension of [39] considered the presence of disturbance, of which a ZMPC that tracks an alternative economic zone is proposed [40]. In both works, the stability theories rely on the assumption that there exists an optimal steady-state operating condition. On the other hand, a generalized ZMPC was proposed with no assumption on steady-state optimal operating point in [41], however without considering any system disturbance or noise.

1.3 Thesis Outline and Contributions

The remaining portion of this thesis is organized as follows:

A comparative study of the performance of three types of representative model approximation methods in MPC and EMPC applications is presented in Chapter 2. To be specific, the three representative types of model approximation methods are 1) model reduction based on available first-principle models, 2) system identification based on input-output data that results in an explicitly expressed mathematical model, and 3) neural networks based on input-output data. Out of each category, we studied the proper orthogonal decomposition (POD) method integrated with trajectory piecewise linearization (TPWL), the subspace identification method, and neural network (NN) training method in the explicit form. The three methods are tested for two processes with very different dynamics, namely an alkylation process and a wastewater treatment plant (WWTP). The MPC and EMPC performance based on each method is compared for each process of interest. The strengths and drawbacks of each method are summarized according to the simulation results.

In Chapter 3, we took the observation obtained in Chapter 2 and investigated further with

the NN training model approximation approach. A case-specific model approximation approach is developed for an agro-hydrological system. The dynamics of the agro-hydrological system are studied and a two-layer NN framework is developed. To minimize the prediction error, a linear bias correction is added to the proposed model. The model is employed by a model predictive controller with zone-tracking (ZMPC), which aims to keep the root zone soil moisture in the target zone while minimizing the total amount of irrigation. The performance of the proposed approximation model framework is shown to be better compared to a benchmark long-short-term-memory (LSTM) model for both open-loop and closed-loop applications. Significant computational cost reduction of the ZMPC is achieved with the proposed framework. To handle the tracking offset caused by the plant-model-mismatch of the proposed NN framework, a shrinking target zone is proposed for the ZMPC. Different hyper-parameters of the shrinking zone in the presence of noise and weather disturbances are investigated, of which the control performance is compared to a ZMPC with a time-invariant target zone.

In Chapter 4, we continued our study with MPC with zone-tracking and focused more on the stability and robustness of the controller. We proposed a robust nonlinear model predictive control design with generalized zone-tracking (ZMPC). The proposed ZMPC has guaranteed convergence into the target zone in the presence of bounded disturbance. The proposed approach achieves this by modifying the actual target zone such that the effect of disturbances is rejected. A control invariant set (CIS) inside the modified target zone is used as the terminal set, which ensures the closed-loop stability of the proposed controller. Detailed closed-loop stability analysis is presented. Simulation studies based on a continuous stirred tank reactor (CSTR) are performed to validate the effectiveness of the proposed ZMPC.

In Chapter 5, we extend the robust ZMPC proposed in Chapter 4 and apply it to a Post-Combustion Carbon Capture (PCC) plant. We modified the terminal constraint to track only the system output instead of the system state to improve the applicability of the controller on

complex large-scale systems. Furthermore, an algorithm that projects the ellipsoidal control invariant set defined in the state space into the output space is developed and presented. The performance of the modified robust ZMPC is validated through simulations and is compared with that obtained based on a benchmark ZMPC controller. The controller is shown to be more robust in the presence of system disturbances, where the flow rate of the flue gas coming into the plant varies over time.

Finally, Chapter 6 summarizes the contribution of this thesis, followed by discussions on potential research directions.

Chapter 2

A comparative study of model approximation methods applied to economic MPC

We perform a critical examination of the three kinds of model approximation methods introduced in Chapter 1 when used in conventional tracking MPC and EMPC in this chapter. One representative model approximation method is selected from each of the three model approximation categories. Specifically, POD integrated with trajectory piecewise linearization (TPWL) [20], subspace identification [21], and explicitly-expressed fully connected neural network are selected. Two processes that are very different in dynamic nature and complexity are selected as benchmark processes. Conventional tracking MPC controllers and EMPC controllers designed based on the three model approximation methods are applied to the benchmark processes. The performance of the control frameworks is compared from different aspects, including computational complexity, tracking performance for MPC, and economic performance for EMPC[1].

Briefly, this chapter is structured as follows. Section 2.1 presents some preliminaries

[1]This chapter has been published as a journal paper.

including the EMPC design employed. In Section 2.2, the selected model approximation algorithms are introduced. Then in Section 2.4 the WWTP process is introduced, followed by the simulation results and some remarks. Finally, Section 2.5 summarizes the work performed in this chapter.

2.1 Preliminaries

2.1.1 System description and problem formulation

In this chapter, we consider model approximation of nonlinear dynamical systems that can be described using the following ordinary differential equations:

$$\dot{x}(t) = f(x(t), u(t)) \tag{2.1a}$$

$$y(t) = g(x(t), u(t)) \tag{2.1b}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^r$ is the input vector, $y \in \mathbb{R}^l$ is the output vector, f and g are nonlinear vector functions, f describes the dynamics of the system and g describes the relation of the output to the state and the input. For convenience, we will refer to the system in (2.1) as the *original system* in the remainder of this chapter.

Model approximation will be applied to either reduce the original system to a system of a lower order that approximates the dynamics of the original system with sufficient accuracy or to identify a linear or neural network model based on the input-output data of the original system. These reduced or identified models will be applied in advanced control frameworks as system constraints and their applicability and performance will be carefully examined. To be specific, conventional set-point tracking MPC and EMPC will be investigated and differences in control behaviors of these two frameworks with respect to different reduced or identified models are of our interest.

2.1.2 Model predictive control design

We consider a MPC formulation that tracks an optimal steady-state operating point (y_s, u_s). The tracking MPC at a specific sampling time t_i is formulated as an optimization problem as follows:

$$\min_{u(t)\in S(\triangle)} \sum_{j=i}^{i+N-1} \Big(|y(t_j|t_i) - y_s|_Q^2 + |u(t_j|t_i) - u_s|_R^2 \Big) + |y(t_{k+N}|t_i) - y_s|_{P_f}^2 \tag{2.2a}$$

$$\text{s.t.} : \dot{x}(t) = f(x(t), u(t)) \tag{2.2b}$$

$$y(t) = g(x(t), u(t)) \tag{2.2c}$$

$$x(t_i) = \bar{x}(t_i) \tag{2.2d}$$

$$u(t) \in \mathbb{U} \tag{2.2e}$$

$$y(t) \in \mathbb{Y} \tag{2.2f}$$

where $S(\triangle)$ is the family of piece-wise constant function with $\triangle$ being the sampling time, N is the prediction horizon and is a finite positive integer, $y(t|t_i)$, and $u(t|t_i)$ represent the prediction of the variables at future time t made at the current time t_i. Q, R and P_f are diagonal weighting matrices, $\bar{x}(t_i)$ is the estimated or measured state vector at current time t_i, $\mathbb{U} \subset \mathbb{R}^r$, and $\mathbb{Y} \subset \mathbb{R}^l$ are compact sets. The optimization problem aims to optimize $u(t)$ from t_i to t_{i+N-1}, such that the accumulated difference between $(y(t|t_i), u(t|t_i))$ and the set-point (y_s, u_s) is minimized, which is defined by the objective function (2.2a). System model (2.2b) - (2.2c) are employed to predict $y(t|t_i)$ with initial condition specified by (2.2d). (2.2e)-(2.2f) are input and output constraints that have to be satisfied over the entire horizon N while solving the problem.

Note that when a reduced or identified model is used to design the MPC controller, the model equations (2.2b)-(2.2c) should be replaced with the reduced or identified model equations accordingly.

The optimal solution of the above optimization problem can be denoted as $u^{\star}_{\text{EMPC}}(t|t_i)$.

However, only the values at the first time instant (i.e. the current time t_i) will be applied to the system:

$$u(t) = u^{\star}_{\mathrm{EMPC}}(t|t_i), \quad t \in [t_i, t_{i+1}) \tag{2.3}$$

Regarding the optimal operating point (y_s, u_s) employed in (2.2a), it may be calculated through a steady-state optimization problem as shown in (2.4),

$$(y_s, u_s) = \min_{y,u} \; l_e(y, u) \tag{2.4a}$$

$$\text{s.t.} \; : \dot{x} = f(x, u) \tag{2.4b}$$

$$y = g(x, u) \tag{2.4c}$$

$$u \in \mathbb{U} \tag{2.4d}$$

$$y \in \mathbb{Y} \tag{2.4e}$$

where the objective function $l_e(y, u)$ represents an economic performance of the system.

2.1.3 Economic model predictive control design

As have presented in the previous section, MPC takes a two-layered design. The optimal operating point is first obtained by solving a steady-state optimization problem. A series of dynamic optimization problems that tracks the states and inputs to the optimal operating point is then proceeded along the horizon of interest. In EMPC design, these two steps are combined into a single step. The objective function used in the steady-state optimization, which is often times economic performance related, is directly employed in the dynamic optimization problems.

A basic EMPC formulation for the system described in (2.1) is employed in this chapter. EMPC is in general formulated as an optimization problem, which takes the following form

at the sampling time t_i [4]:

$$\min_{u(t)\in S(\triangle)} \sum_{j=i}^{i+N-1} l_e(y(t_j|t_i), u(t_j|t_i)) \tag{2.5a}$$

$$\text{s.t.} : \dot{x}(t) = f(x(t), u(t)) \tag{2.5b}$$

$$y(t) = g(x(t), u(t)) \tag{2.5c}$$

$$x(t_i) = \bar{x}(t_i) \tag{2.5d}$$

$$u(t) \in \mathbb{U} \tag{2.5e}$$

$$y(t) \in \mathbb{Y} \tag{2.5f}$$

The same definitions as used in the MPC formulation are also valid for the EMPC formulation. The only difference lies in the objective function as used in (2.5a), where the economic performance index employed in the steady-state optimization is adopted in dynamic optimization directly.

2.2 Model Approximation Methods

In this section, the model approximation methods encountered in this chapter are briefly introduced. The characteristics of each approaches are discussed, followed by step-by-step descriptions of how the methods can be implemented.

2.2.1 Model reduction based on first-principle model

First, we introduce the proper orthogonal decomposition (POD) and trajectory piecewise linearization (TPWL) methods. The two methods represent the two popular approaches to reduce model complexity respectively, namely reducing the model order and mimicking complex mathematical relationships in simpler structures (e.g. linearization). Applications of these two methods can be found in various works, for example, [20, 42] for POD and [20, 43] for TPWL. A combination of POD and trajectory piecewise linearization (TPWL)

is considered in this chapter.

The key idea of the POD method is to obtain a projection matrix that projects the original state vector onto a lower-dimensional space. To achieve this goal, a snapshot matrix of the states X is first generated by simulating the original system with an input trajectory that ensures persistent excitation of the system. The snapshot X contains sampled system states at different time instants. Suppose that:

$$X = [x(t_1), x(t_2), x(t_3), \cdots, x(t_N)] \in \mathbb{R}^{n \times N}$$

where $x(t_i)$, $i = 1, \cdots, N$, is the sampled state vector at time t_i, N is the total number of samples, and n is the dimension of the state vector. Note that N should be much larger than n to ensure that the snapshot captures the dynamics of the original system. After X is generated, singular value decomposition (SVD) is applied to X to obtain the following decomposition:

$$X = U\Sigma V^{\mathrm{T}} \tag{2.6}$$

where $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{N \times N}$ are orthogonal matrices and are referred to as the left and right singular matrix of X respectively, and $\Sigma = \mathrm{diag}^1([\sigma_1, \sigma_2, \cdots, \sigma_n]) \in \mathbb{R}^{n \times N}$ is a diagonal matrix, where $\sigma_1, \cdots, \sigma_n$ are the n singular values of X and are arranged in a descending order. It is noted that the larger the value of a singular value, the more important system dynamics it captures. In POD, the order of the reduced system is determined based on these singular values. It is recommended that the order k is obtained such that there is a significant drop in the singular value magnitude from σ_k to σ_{k+1}. Once k is determined, the projection matrix $U_k \in \mathbb{R}^{n \times k}$ is defined to be the first k columns of the matrix U. The reduced system state $z(t)$ has the following relationship with the original state $x(t)$:

$$x(t) = U_k z(t) \tag{2.7}$$

[1] diag(v) denotes a diagonal matrix with the diagonal elements being the elements of the vector v

It is to be noted that although POD reduces the system order to a smaller number, the underlying calculations in general do not reduce from a numeric optimization point of view. This is because that it is in general challenging to explicitly express the reduced system for medium to large-scale nonlinear systems. Thus when the original system has complex nonlinear dynamics, applying only POD to the system often does not reduce the computational complexity. A common way to deal with this problem is to simplify the mathematical complexity of the original system first, followed by applying POD on the simplified model. Readers can refer to [20] for an example application of this approach, where POD was applied to a linear time-varying (LPV) reduced model obtained using TPWL.

Instead of linearizing at a single point, in TPWL, the original system is linearized along a typical state trajectory at multiple points. With a predetermined number of linearization points, the operating region of interest should be covered evenly by the selected linearization points. This can be achieved by the application of a distance-based algorithm which is introduced in [44]. After linearization, the linearized models at different points are combined through weighted summation to form an LPV model. The weighting function may be designed in a way such that once the state-input pair approaches one of the linearization points, the weight for the corresponding sub-model increases to one quickly.

Let us assume that there are s linearization points. The weighting function $w_i(\cdot), i = 0, \cdots, n-1$ that combines the linearized models can be determined following [44]. The procedure for implementing POD with TPWL is summarized as follows[44]:

1. Run open-loop simulations using the original model and generate the snapshot matrix X using the open-loop simulation data.

2. Based on the snapshot matrix X, calculate the projection matrix U_k as the truncated left singular matrix U_k of the snapshot matrix X following (2.6) and (2.7).

3. Based on extensive simulations of the original system, determine a representative trajectory of the system and along the trajectory, determine the s linearization points

based on the distance-dependent algorithm discussed in [44].

4. Obtain the s linearized models by linearizing the original system at each of the s linearization points.

5. Based on the s linearized models and the weighting function $w_i(\cdot), i = 0, \cdots, n-1$, the LPV model that combines all the linearized models takes the following form:

$$\dot{x} = \sum_{i=0}^{s-1} w_i(x)(A_i(x - x_i) + B_i(u - u_i) + f(x_i, u_i)) \tag{2.8a}$$

$$y = \sum_{i=0}^{s-1} w_i(x)(C_i(x - x_i) + D_i(u - u_i) + g(x_i, u_i)) \tag{2.8b}$$

where x_i, u_i for $i = 0, 1, ...s-1$ represent the linearization points, and A_i, B_i, C_i, D_i are the partial derivatives $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial u}$ and $\frac{\partial g}{\partial x}$, $\frac{\partial g}{\partial u}$ evaluated at the linearization points, respectively.

6. Use the projection matrix U_k to project the high-order LPV model (2.8) onto a lower-order state-space as follows:

$$\dot{z} = \sum_{i=0}^{s-1} w_i(U_k z)U_k^T[(A_i(U_k z - U_k z_i) + B_i(u - u_i) + f(U_k z_i, u_i))] \tag{2.9a}$$

$$y = \sum_{i=0}^{s-1} w_i(U_k z)(C_i(U_k z - U_k z_i) + D_i(u - u_i) + g(U_k z_i, u_i)) \tag{2.9b}$$

where z_i represents the ith reduced linearization point.

2.2.2 Subspace model identification

When the first-principle model is not available, model identification based on input-output data of the system is the common approach for model development. There are many different model identification algorithms. In this chapter, we will focus on subspace identification. Unlike other traditional identification methods that require predetermined model structure

and order, subspace identification has the unique property that the model can be identified with only the input-output dataset. Furthermore, the subspace method is purely linear-algebra-based and does not involve solving optimization problems [18].

The subspace identification method leads to a discrete-time linear time-invariant (LTI) state-space model in the following form:

$$z_{k+1} = Az_k + Bu_k, \tag{2.10a}$$

$$y_k = Cz_k + Du_k, \tag{2.10b}$$

where $z_k \in \mathbb{R}^s$, $u_k \in \mathbb{R}^r$ and $y_k \in \mathbb{R}^l$ are the state vector, input vector and output vector, respectively, $A \in \mathbb{R}^{s\times s}$, $B \in \mathbb{R}^{s\times r}$, $C \in \mathbb{R}^{l\times s}$ and $D \in \mathbb{R}^{l\times r}$ are system matrices. The model is obtained by first calculating the extended observability matrix of the system using techniques such as oblique projection, from which matrices C and A can be calculated. Matrices B and D can then be obtained using least square methods.

One type of the subspace identification algorithms can be summarized as follows[18]:

1. Calculate the oblique projection of Y_f along the row space of U_f on to the row space of Z_p, where Y_f, U_f, and Z_p are the block Hankel matrix consists of future outputs, future inputs and past inputs and outputs ($Z_p = [U_p^T \ Y_p^T]^T$, U_p, and Y_p are the Hankel matrix consist the past input and outputs) respectively.

$$O_i = Y_f /_{U_f} Z_p \tag{2.11}$$

2. Perform SVD on O_i:

$$O_i = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^{\mathrm{T}} \\ V_2^{\mathrm{T}} \end{pmatrix}, \tag{2.12a}$$

$$\Gamma_i = U_1 S_1^{1/2}, \tag{2.12b}$$

$$X_f = \Gamma_i^{\dagger} O_i, \tag{2.12c}$$

where the columns of $(U_1 \quad U_2)$ are the left singular vectors of O_i, S_1 is the diagonal matrix that contains the nonzero singular values of O_i, Γ_i is the extended observability matrix, X_f is the Hankel matrix that consists the future states, and $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of the matrix $(\cdot)$. The dimension s of the state is determined by the dimension of S_1.

3. Compute the extended observability matrix Γ_i based on (2.12b), and the orthogonal complement of the row space of Γ_i can be calculated by $\Gamma_i^{\perp} = U_2^{\mathrm{T}}$, where $(\cdot)^{\perp}$ denotes the orthogonal complement of the row space of the matrix $(\cdot)$.

4. Determine C as the first l rows of Γ_i, where l is the dimension of the output vector. According to the definition of Γ_i, we can find the following relation:

$$\begin{pmatrix} C \\ CA \\ CA^2 \\ \cdots \\ CA^{i-2} \end{pmatrix} A = \begin{pmatrix} CA \\ CA^2 \\ CA^3 \\ \cdots \\ CA^{i-1} \end{pmatrix}, \quad \underline{\Gamma_i} A = \overline{\Gamma_i},$$

where $\underline{\Gamma_i}$ denotes Γ_i without the last l rows, and $\overline{\Gamma_i}$ is Γ_i without the first l rows. Then A can be solved by $A{=}\underline{\Gamma_i}^{\dagger}\ \overline{\Gamma_i}$.

5. Calculate the matrix $\Gamma_i^{\perp} Y_f U_f^{\dagger}$, and construct

$$\Gamma_i^{\perp} Y_f U_f^{\dagger} = \begin{pmatrix} M_1 & M_2 & \cdots & M_i \end{pmatrix},$$
$$\Gamma_i^{\perp} = \begin{pmatrix} L_1 & L_2 & \cdots & L_i \end{pmatrix},$$

then B and D can be solved from

$$\begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_i \end{pmatrix} = \begin{pmatrix} L_1 & L_2 & \cdots & L_{i-1} & l_i \\ L_2 & L_3 & \cdots & L_i & 0 \\ L_3 & L_4 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ L_i & 0 & \cdots & 0 & 0 \end{pmatrix} \begin{pmatrix} I_l & 0 \\ 0 & \underline{\Gamma_i} \end{pmatrix} \begin{pmatrix} B \\ D \end{pmatrix}$$

2.2.3 Neural network modeling

The last method to be introduced is to train neural networks (NNs) as system models. Similar to the subspace identification method introduced in the previous section, neural network training does not require a significant amount of knowledge regarding the dynamics of the interested system. Furthermore, NNs can capture nonlinear dynamics while the subspace identification is a linear method. According to the universal approximation theorem [45], neural networks in general are capable of approximating any function $f : \mathbb{R}^n \to \mathbb{R}^m$ with n-dimension inputs and m-dimension outputs up to some maximum error ε. In this chapter, fully connected neural networks with nonlinear activation functions are considered. Unlike models obtained with traditional identification methods, neural networks are often used as black-box models without explicit expressions, which could be challenging for advanced control applications. This is because many of the conventional optimization solvers require Jacobian/Hessian information of the systems, which could be computationally expensive for black-box models. In this chapter, NN system models are investigated in the form of basic

black-boxes as well as explicitly expressed systems of equations.

A brief step-by-step procedure for identifying a NN model is outlined as follows [46]:

1. Determine the dimension of the NN input and output vectors:

$$NN_{in} \in \mathbb{R}^{N_{past}\cdot(r+l)+(N_{future}-1)\cdot r}$$
$$NN_{out} \in \mathbb{R}^{N_{future}\cdot l}$$

   where N_{past} is the number of past steps considered, and N_{future} is the number of future time steps to be predicted. Note that the inputs of the NN includes both system inputs and outputs from the past. Furthermore, when system outputs at multiple future steps are of interests, the future system inputs for the corresponding steps are as well required as NN inputs.

2. Determine the number of NN layers, the number of nodes in each layer, and the activation function to be used. Note that there are no universal rules for determining these parameters. Users can manually do guess-and-check or use tools such as grid search.

3. Reconstruct the dataset according to the predetermined input-output dimension.

4. Identify the NN using computational tools (e.g. the state-of-the-art application program interface (API) Keras [47]).

5. Extract the trained parameters from the model.

6. Define explicit models with the trained parameters and the known NN structure. A single fully-connected NN can be defined as follows:

$$y = \sigma(\sum_{i=1}^{n_{node}} w_i x_i + b_i) \tag{2.13}$$

   where x and y represents the input and output of the NN layer, w_i and b_i are the weighting factor and the bias for the i^{th} input, n_{node} is the number of nodes in the NN

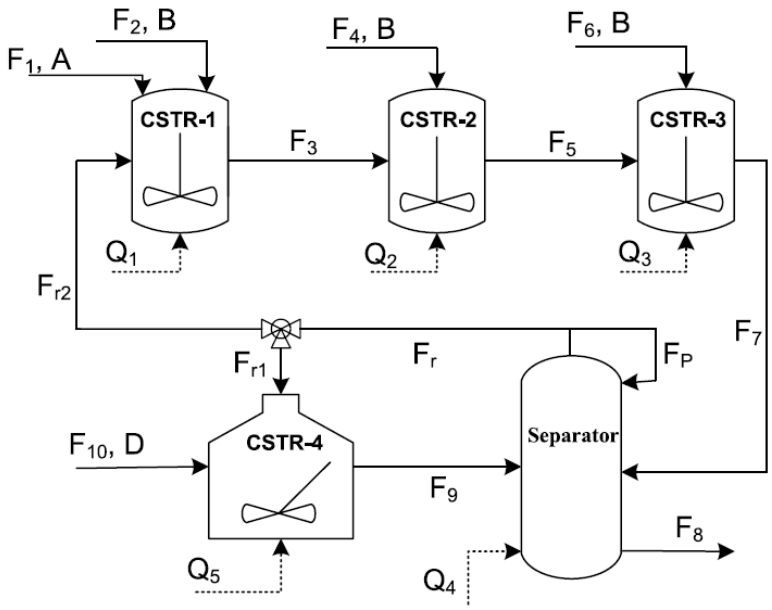

Figure 2.1: A schematic of the alkylation of benzene process

layer, $\sigma(\cdot)$ is the activation function.

In the following two sections, we consider two benchmark process examples and study the performance of the three model approximation methods when they are used in MPC and EMPC designs. The two benchmark processes are selected to be very different in scales and dynamic natures, such that the performance of the model approximation methods applied to different types of systems can be investigated and a more comprehensive comparison can be made.

2.3 Case Study 1: Benzene Alkylation Process

2.3.1 Process overview

The first benchmark process considered is a benzene alkylation process. A schematic diagram of the process is presented in Figure 2.1 [48]. The process considered consists of four CSTRs and a flash tank separator. Stream F_1 consist of pure benzene while streams F_2, F_4 and F_6 consist pure ethylene. Two catalytic reactions take place in CSTR-1, CSTR-2, and

CSTR-3, of which benzene (A) react with ethylene (B) and produce the desired product ethylbenzene (C) (reaction 1), while ethylbenzene further react with ethylene and produce the byproduct 1,3-diethylbenzene (D) (reaction 2). The effluent of CSTR-3 is fed to the flash tank separator, where the desired product is separated and leave from the bottom stream F_8. On the other hand, the left-over benzene leaves the separator from the top and splits into F_{r1} and F_{r2} which go into CSTR-4 and CSTR-1 respectively. 1,3-diethylbenzene is fed into CSTR-4 through F_{10}, where it reacts with benzene to produce ethylbenzene under the presence of catalysts (reaction 3). All the chemicals leaving from CSTR-4 eventually go to the separator. Each vessel has an external heat supply stream which helps to adjust the vessel temperature (Q_1 - Q_5).

The state variables of the process are the concentrations of the four chemical components and the temperature in each vessel. The manipulated inputs are the external heat supply and the feed stream flow rates of ethylene going into CSTR-2 and CSTR-3. Overall, the system consists of 25 states and 7 manipulated inputs. A more detailed model description with parameter values can be found in [48].

2.3.2 Control objective and controller setting

The control objective is to maximize the production rate of ethylbenzene while keeping the energy cost at a reasonable level. Thus the economic performance measure employed for the steady-state optimization and the EMPC controller is defined as follows:

$$l_e = -\alpha F_8 C_{C4} + \beta \sum_{i=1}^{5} |Q_{is} + Q_i| \tag{2.14}$$

where F_8 and C_{C4} are the flow rate and ethylbenzene concentration leaving the separator respectively, Q_i represents the deviation heat inlet to vessel i with respect to the steady-state values. Q_{is} is the steady-state heat inlet for the corresponding vessel, $\alpha = 200$, and $\beta = 0.0005$ are the weighting coefficients selected so that a fair trade-off between production

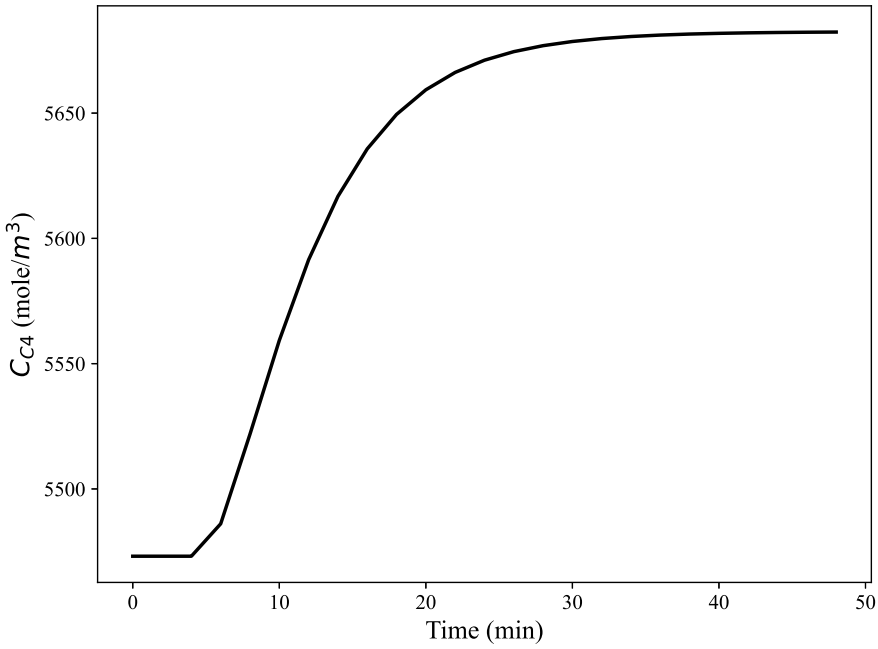

Figure 2.2: Step response of the alkylation process output.

rate and operating cost is established. Note that although the flow rates of ethylene feeding into CSTR-2 and CSTR-3 do not present explicitly in the objection function, the value of F_8 however depends on them.

The weighting factors used in the MPC tracking objective function (2.2a) were selected to be $Q = 1$, $R = \text{diag}([7.5, 5, 5, 6, 5, 4.9 \cdot 10^{-4}, 4.9 \cdot 10^{-4}])$ and $P_f = 10$, where a higher penalty was added to the terminal output to enhance convergence. For simplicity, the output vector y is C_{C4}, which is the only state appeared in (2.14). The observability of the system is confirmed using the PBH test along the data trajectory with the selected output. The optimal steady-state operating point $y_s = 5526$, $u_s = [1.0, 1.0, 1.0, -1.0, -0.86, 1.0, 1.0]$ is obtained by solving the steady-state optimization (2.4). The input vector and the output

are bounded based on physical properties of the process.

$$\mathbb{U} = \{u|\, \underline{u} \leq u \leq \bar{u}\} \tag{2.15a}$$

$$\mathbb{Y} = \{y|\, y \geq 0\} \tag{2.15b}$$

where $\underline{u}$ and $\bar{u}$ are the upper and lower bound of the input vector. The exact values of $\underline{u}$ and $\bar{u}$ can be found in [48].

The controller sampling time is 2 minutes, and the prediction horizon is $N = 15$. The two parameters are picked based on the output response (Figure 2.2) with a unit step change in the input vector u. As presented in Figure 2.2, the major dynamics of the system can be captured in the 30-minute window defined by the sampling time and prediction horizon.

2.3.3 Data generation and model approximations

All the methods discussed in section 2.2 require the generation of a dataset that can represent the dynamics of the original process. Given that the considered alkylation process is nonlinear, multi-level input signals are used as the input excitation to the process for input-output data generation. Open-loop simulations are carried out with the multi-level inputs applied to the original nominal system model. For consistency, one dataset is generated and is used for all the model approximation methods.

Figure 2.3 shows a portion of the input signal employed for the alkylation process and the corresponding output response. Note that the input signal is normalized with respect to the steady-state.

At a given time step k, each element in the input vector u randomly takes a value from a predetermined set $\mathbb{U}_d$ and holds onto that value until time $(k + n_{hold})$. Note that $n_{hold} \in \mathbb{I}_a^b = \{a, a+1, ..., b\}$, where $\mathbb{I}_a^b$ is an integer set with a and b being the lower and upper bounds. At time step $k + n_{hold}$, the element values of the input vector u will be reselected from $\mathbb{U}_d$.

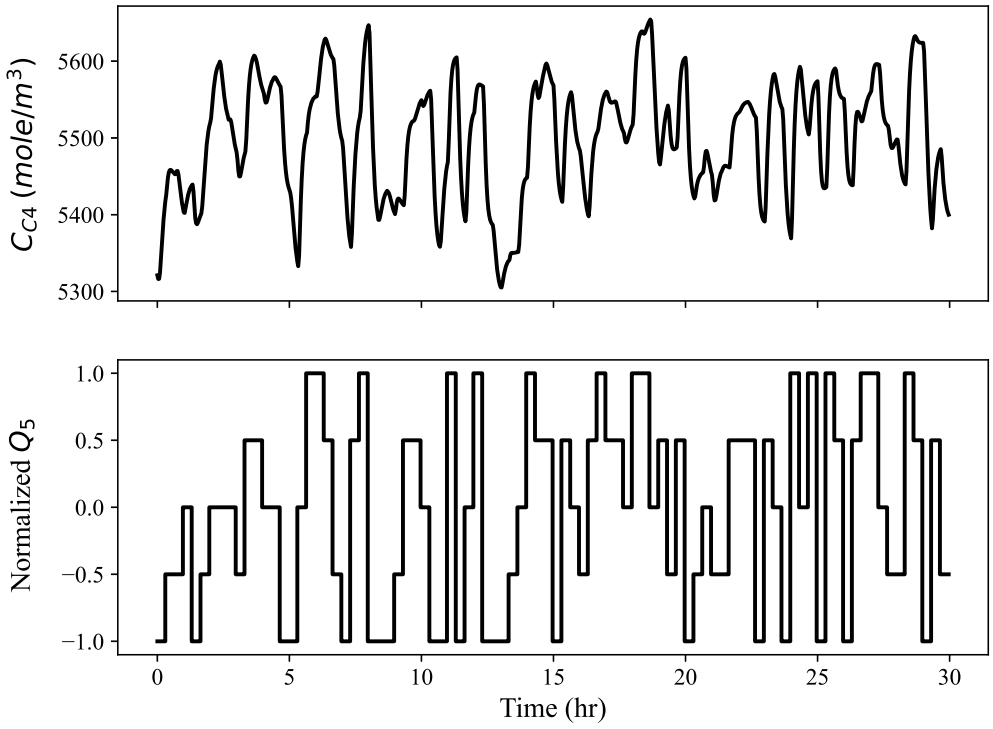

Figure 2.3: Multi-level input and the corresponding output response.

The sets $\mathbb{U}_d$ and $\mathbb{I}_a^b$ can be designed based on the dynamics of the system such that the frequency and magnitude of the step changes can excite the system properly. If only the upper and lower bounds of the input vectors are included in $\mathbb{U}_d$, the input signal becomes a pseudorandom binary sequence (PRBS). For the alkylation process to be specific, the set $\mathbb{U}_d$ was designed to have five elements which were obtained by equally splitting the input domain $\mathbb{U}$:

$$\mathbb{U}_d = \{\underline{\mathrm{u}}, \underline{\mathrm{u}} + \frac{\delta_u}{4}, \underline{\mathrm{u}} + \frac{\delta_u}{2}, \underline{\mathrm{u}} + \frac{3\delta_u}{4}, \bar{u}\},\ \delta_u = \bar{u} - \underline{\mathrm{u}}$$

Taking into account of the rapid ongoing chemical reactions in the alkylation process, the integer set $\mathbb{I}_a^b$, $a = 10, b = 16$ was used. This means that the input will hold onto the same values for 20 to 32 minutes before the next set of values can be picked from $\mathbb{U}_d$.

To ensure reasonable results from the considered methods, 5000 data points were gener-

ated. The reduced model order k was determined by guess and check, where reduced models of different orders were generated based on the original model first and open-loop prediction performances were checked and compared through a step test. The model order that gives a satisfying balance between prediction accuracy and model order was selected and employed for the TPWL-POD models.

Data scaling and reconstructions are required for NN identification such that the pre-selected input-output dimension is satisfied. The Python library Keras [47] was used to identify a fully-connected NN model which takes the form of:

$$f : \mathbb{R}^{359} \to \mathbb{R}^{15}$$

where $359 = 30 \times 8 + 14 \times 7$, 8 is the number of inputs plus outputs, 30 is the number of past data points used, 7 is the number of inputs, and 14 is the number of future inputs encountered. The model takes 30 points from the past plus 17 future inputs and predicts system outputs 15 steps into the future (including the current step). Note that the number of predictions obtained by the model is the same as the length of the prediction horizon($N = 15$), meaning that the NN needs to be called only once in the optimization problem. This helps to reduce the prediction offset caused by identifying a single-step-ahead prediction NN and calling it recursively. As for activation functions, `'sigmoid'` and `'swish'` were used in this chapter. Note that `'sigmoid'` function leads to smooth nonlinear NN dynamics, while `'swish'` captures steep changes better but is more aggressive at the same time.

All simulations were done in Python. Different programming libraries were employed for different model approximation methods. To be specific, MPCTools [49] was used for solving TPWL/POD-model-based EMPC, and CasADi [50] was used for solving subspace-model-based EMPC and NN-model-based EMPC. Note that the optimization solver IPOPT [51] was employed for all EMPC simulations.

2.3.4 Results and discussion

2.3.4.1 Approximation models

Six models are developed to approximate the nonlinear dynamics of the process. First, two TPWL models are developed by linearizing the original nonlinear model around 3 and 5 points respectively. Then, POD method is applied to the TPWL models, leading to two POD-TPWL models. One LTI model that takes the form of (2.10) is developed using the subspace identification method. The dimension of the LTI model is 2. A fully-connected NN with two hidden layers is generated. The hidden layers have 10 and 3 nodes individually.

The original model is used as the reference for performance evaluation of the approximation models. The output responses and the corresponding input trajectory of the original model and the approximation models are shown in Figure 2.4. The prediction performance is measured by the normalized root mean square error (NRMSE):

$$NRMSE = \frac{1}{y_s}\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{2.16}$$

where y_s is the optimal steady-state output, n is the number of data points, $\hat{y}_i$ is the ith output predicted by the approximation model, and y_i is the ith output obtained from the original nonlinear model. The NRMSEs of the approximation models are summarized in Table 2.1.

From Figure 2.4, it is observed that the TPWL and POD-TPWL models with five linearization points perform better than models with three linearization points. The application of the POD method did not impact the prediction performance significantly. Table 2.1 also shows consistent results. It was noticed that models perform differently with different variations in inputs. The NN model has a more noisy response compare to other models. The reason for this is that NNs are complex nonlinear functions with a significant number of parameters. Thus it is impractical to obtain the absolute optimal set of parameters, making

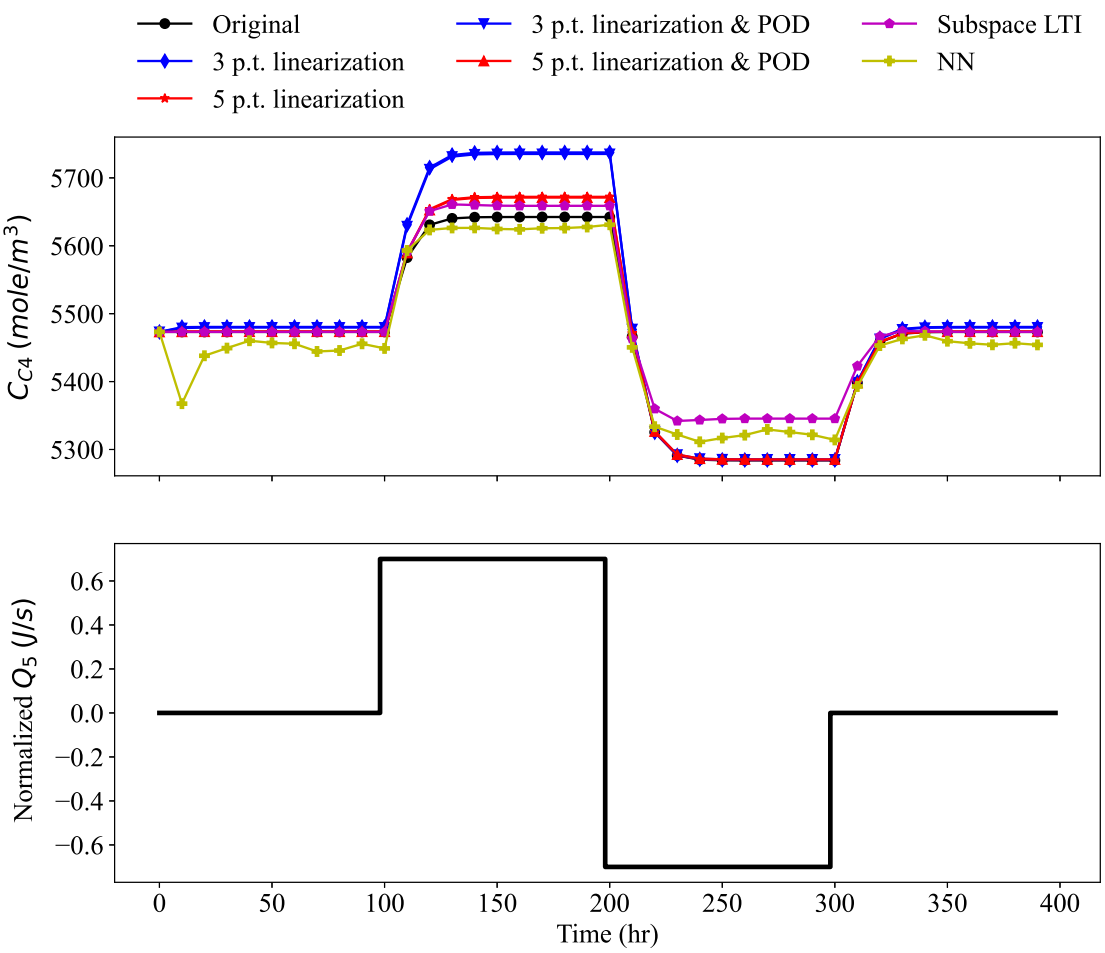

Figure 2.4: Open-loop output responses of the alkylation first-principle model and approximation models.

numerical prediction errors harder to eliminate. Furthermore, it is noteworthy that NNs works with scaled variables in the range of -1 to 1, leading to more numerical errors. All the models respond to input changes without delays and converge to values that are very close to the steady-state output of the original nonlinear system when the input returns back to the steady-state value.

2.3.4.2 Conventional tracking model predictive control

The approximation models discussed in the previous section are employed by the MPC controller (2.2) with parameter settings as discussed in Section 2.3.2. The simulation results

Table 2.1: Open-loop prediction performance of approximation models - Alkylation

Model applied	Dimension	Normalized RMSE
3 p.t. linearized	25	0.008
5 p.t. linearized	25	0.002
3 p.t. linearized & POD	15	0.008
5 p.t. linearized & POD	15	0.002
Subspace LTI	2	0.005
Neural Network	N/A	0.005

are summarized in Table 2.2. The order of the reduced model, the average time consumed in solving the MPC problem at a given time step, and the accumulated objective function value under the corresponding trajectories are presented for the considered models. Figures 2.5 and 2.6 present the tracking performance of the system output (product concentration) and the optimal control strategy of the heat supply going into CSTR-4, which is one of the manipulated inputs. The dashed black line represents the tracking set-point, while the tracking performances obtained with different approximation models are differentiated by colors and markers.

As can be observed in Table 2.2, TPWL models failed to improve the average computational speed for solving the optimization problem but provide satisfying tracking results. This could be caused by the complexity of the TPWL models, leading to more iterations required before the numerical solver converges to the optimal solution. The increased number of linearization points does not improve MPC controller performance. A small offset in the manipulated variable can be observed in Figure 2.6. Computational speed improvements were observed with model order reduction applied on top of TPWL models while the tracking performance became worse. The observation from Figure 2.5 matches the results in Table 2.2, where slight offsets in tracking were observed for the TPWL-POD models. This is

Table 2.2: MPC results of the original model and approximation models - Alkylation

Model applied	Dimension	Time (single step EMPC)	Objective function value
Original	25	1.16	23564.71
3 p.t. linearized	25	4.28	15619.71
5 p.t. linearized	25	2.36	22655.89
3 p.t. linearized & POD	15	0.52	46990.54
5 p.t. linearized & POD	15	0.73	48512.75
Subspace LTI	2	0.07	49933.00
Neural Network	N/A	0.27	182511.67

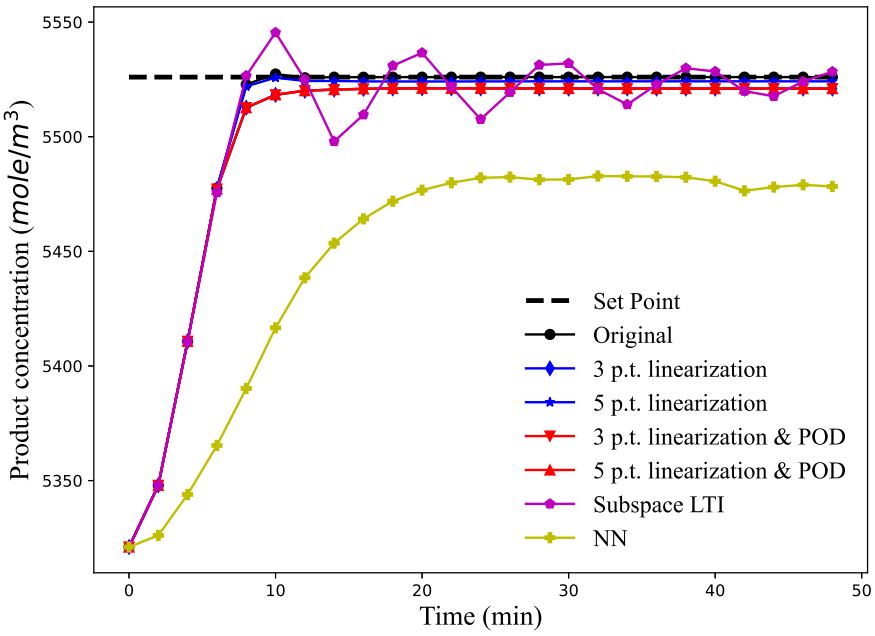

Figure 2.5: The optimal product concentration trajectory under different model-approximation-based conventional MPC frameworks.

reasonable since applying model order reduction and linearization at the same time further reduces the model accuracy.

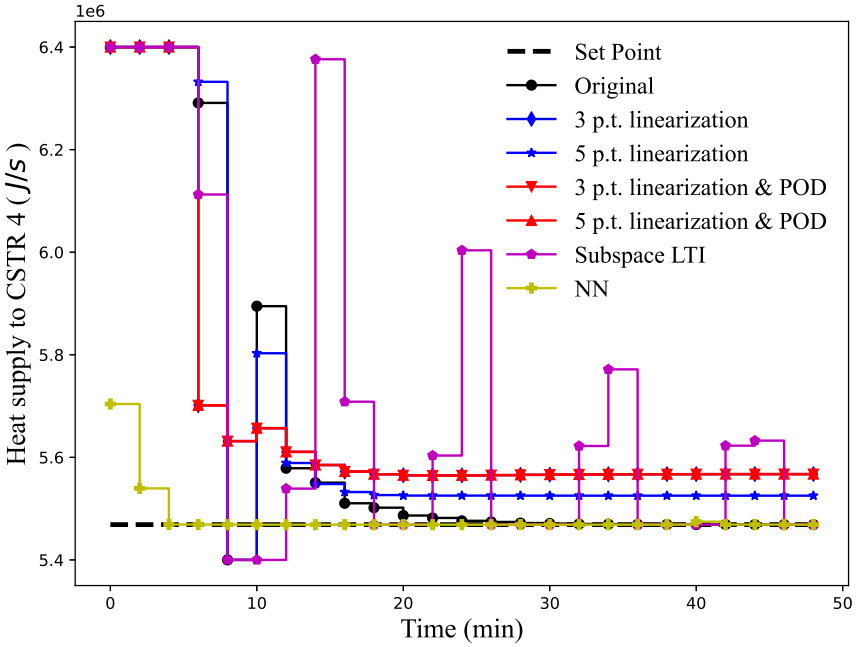

Figure 2.6: Manipulated input trajectory under different model-approximation-based conventional MPC frameworks - the heat supply going into CSTR-4

The subspace-model-based MPC provides significant computational cost reduction with an under-damping response as presented in Figures 2.5 and 2.6. It is also noticed that the control strategy obtained with the subspace model changes rapidly and more aggressively. This observation is reasonable as the subspace model is linear. Compare to nonlinear models that give predictions that are smoother and closer to the actual dynamics of the system, the linear model leads to more aggressive prediction and thus more aggressive controller responses.

As for the NN model, a more significant offset was observed. The offset could be due to the input-output normalization required during training as well as prediction applications, leading to significant numerical errors. The non-smooth noisy prediction of the model also contributed to the offset. Computational cost reduction that is better than the first-principle-model-based models but worse than the subspace model was achieved. Note that the presented input happened to converge to the set-point for the NN model, which is not

Table 2.3: EMPC results of the original model and approximation models - Alkylation

Model applied	Dimension	Time (single step EMPC)	Objective function value
Original	25	0.68	-6867.40
3 p.t. linearized	25	0.67	-4244.25
5 p.t. linearized	25	0.88	-6840.55
3 p.t. linearized & POD	15	0.60	-3012.19
5 p.t. linearized & POD	15	1.39	-6825.65
Subspace LTI	2	0.06	-6739.46
Neural Network	N/A	0.52	-5396.95

the case for all input variables.

2.3.4.3 Economic model predictive control

In this subsection, we study the performance of the different models in EMPC (2.5). The EMPC simulation results of the alkylation process are summarized in Table 2.3. Figure 2.7 presents the product concentration under the optimal control trajectory obtained under different model-approximation-based EMPC frameworks, of which an example is shown in Figure 2.8, where the trajectory of the heat supply for CSTR-4 is presented.

The TPWL model with 5 linearization points gives the best performance. Significant performance improvements are observed with increased linearization points regardless of the presence of model order reduction, which is different from the observation made for MPC controller design. It was also observed that the computational cost rises as the number of linearization points increases. Applying POD on top of TPWL leads to worse EMPC performance, which is similar to what has been observed in the MPC simulation results. When the reference model is relatively simple, further applying POD reduces the computational cost. The opposite is observed with increased reference model complexity (i.e when more lineariza-

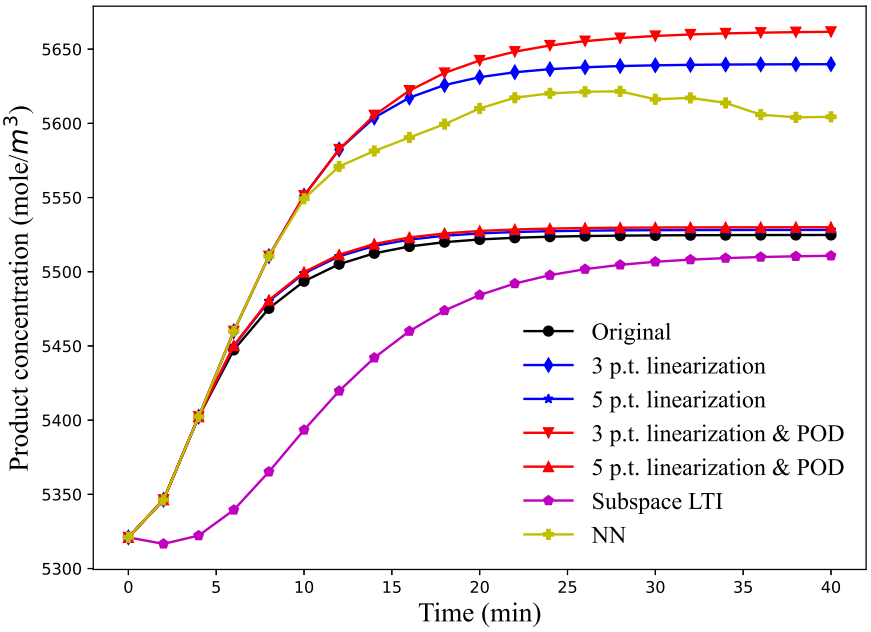

Figure 2.7: The optimal product concentration trajectory under different model-approximation-based EMPC frameworks.

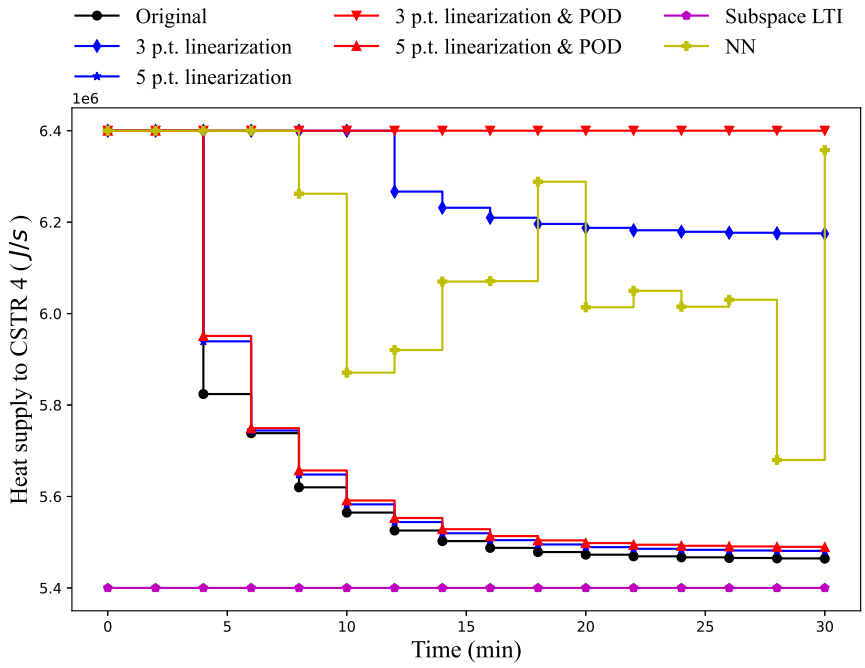

Figure 2.8: Manipulated input trajectory under different model-approximation-based EMPC frameworks - the heat supply going into CSTR-4

tion points are used). Furthermore, the performance deviates more significantly when the referencing LPV model is less accurate. Note that from Figure 2.7, it can be observed that if only the product concentration is considered, different conclusions regarding simulation performance will be obtained, as the heating cost index is ignored. However, it can still be observed from Figure 2.7 and 2.8 that the 5-point-TPWL model gives control strategy that is closest to that obtained with the original model.

Overall, no significant improvement in computational speed is achieved from using first-principle-model-based model reduction. The reason is that the complexity of the original system is relatively low, which can be proved by the fast computational speed while solving the EMPC problem with the original model. In the following section, it will be shown that the speed improvements with the usage of reduced models are significant for the more complex WWTP system.

As can be observed from Table 2.3, the computational load reduction with the subspace model employed is the highest out of all models considered. Two factors contributed to this result, namely the significant reduction in the model order and the linearity of the subspace model. A reasonable EMPC performance was also achieved with the model.

The optimal control trajectory obtained with the NN model is similar to that obtained with the models with 3 linearization points. The response is less smooth and oscillations in the control strategy can be observed from Figure 2.8. The improvement in computational speed achieved is slightly less than the subspace LTI model but is still quite significant. The overall performance of the NN model is better than models with 3 linearization points, but worse than models with 5 linearization points and the subspace LTI model.

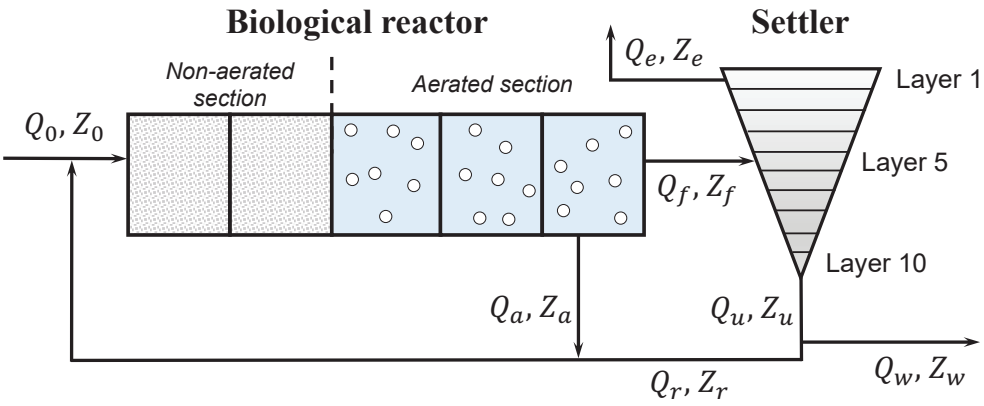

Figure 2.9: A schematic of the wastewater treatment plant.

2.4 Case Study 2: Wastewater Treatment Plant

2.4.1 Process overview

In this section, we consider a wastewater treatment plant which is a more complex nonlinear system. A schematic of the WWTP process is shown in Figure 2.9. The WWTP process consists of an activated sludge bioreactor with five compartments and a secondary settler with ten layers. The reactor consists of two sections, being the non-aerated section including the first two chambers, and the aerated section which includes the latter three chambers. The wastewater stream enters the plant with a concentration Z_0 and a flow rate Q_0. After passing through the reactor, the effluent is separated into two streams, in which one of them is recycled back into the first reactor chamber at flow rate Q_a and the other is fed into the settler on the fifth layer. The stream leaving from the top of the settler contains the purified water that can be disposed, while the bottom effluent is split into a recycle stream with flow rate Q_r and a waste sludge stream Q_w.

A detailed description of the dynamical model of the WWTP can be found in [52]. The model considers 8 biological reactions with 13 major compounds, including 7 soluble compounds and 6 insoluble compounds. The concentrations of the 13 compounds in each of the reactor chambers are the state variables. In the settler, a variable that represents all

insoluble compounds is defined, leading to 8 states for each settler layers. The dynamical first-principle model of the WWTP consists of 145 states. Two manipulated input variables are considered, namely the oxygen transfer rate in the fifth chamber of the reactor KL_{a5} and the flow rate of the recirculation stream Q_a.

2.4.2 Control objective and controller setting

The economic control objective used for the EMPC controller and the steady-state optimization layer of MPC are adopted from [15]. Two performance indices are considered for assessing the economic performance of the process, namely effluent quality (EQ) and overall cost index (OCI). The economic performance measure is a weighted summation of the two performance indices as follows:

$$l_e = \alpha \cdot \mathrm{EQ} + \beta \cdot \mathrm{OCI} \tag{2.17}$$

where EQ (kg pollution $\cdot$ day^{-1}) is a factor that measures the processed water quality evaluated as the daily average of a weighted summation of various effluent compound concentration, OCI covers the factors that have significant impacts on the process operating cost including sludge production (kg $\cdot$ day^{-1}), aeration energy (kWh $\cdot$ day^{-1}), pumping energy (kWh $\cdot$ day^{-1}), and mixing energy (kWh $\cdot$ day^{-1}), and α and β are two weighting factors for EQ and OCI, respectively. More detailed description of the cost function can be found in [15].

The system outputs are chosen to be the states required to calculate EQ and OCI. The observability of the system is validated by PBH test. In total, 41 outputs are used. However, with only 2 inputs, it is very challenging and in fact impossible to track all 41 outputs. Thus the MPC controller is designed to track two of the outputs that deviate the most from the set-point, namely the slowly biodegradable and soluble substrate in the first and second reactor compartments. The two tracked outputs will be denoted as XS_1 and XS_2

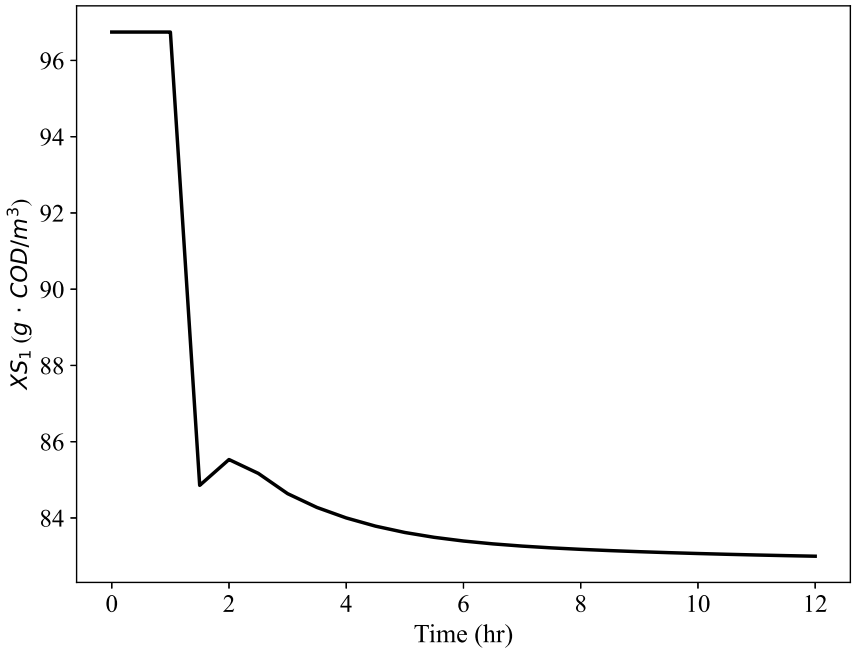

Figure 2.10: Step response of a WWTP process output.

in the following sections. The weighting parameters are defined to be $Q = diag([100, 100])$, $R = diag([100, 100])$ and $P_f = diag([1000, 1000])$. Similar to the alkylation process, the input and output vectors are only bounded:

$$\mathbb{U} = \{u |\ \underline{u} \leq u \leq \bar{u}\} \tag{2.18a}$$

$$\mathbb{Y} = \{y |\ y \geq 0\} \tag{2.18b}$$

where $\underline{u} = [0, 0], \bar{u} = [240, 92230]$. Scaling is performed in implementations to reduce numerical errors.

An open-loop response of XS_1 with respect to step changes in the input vector at $t = 1\,hr$ is presented in Figure 2.10. The controller sampling time is 30 minutes and the prediction horizon is $N = 10$. The dynamics five hours into the future are considered by the controller, which is enough to cover the essential dynamics of the process according to Figure 2.10.

2.4.3 Data generation and model approximation

A dataset with 5376 data points (112 simulation days) is generated for the WWTP process. Samples of the training dataset are presented in Figure 2.11. Two output trajectories and one input trajectory are presented. Recall that XS_1 is the slowly biodegradable and soluble substrate in the first reactor compartment, Q_a is the flow rate of the recirculation stream. X_1 represents the total sludge concentration in the bottom layer of the settler. Note that the input signal is designed differently compare to the alkylation process. Instead of equally splitting the input domain, the signal is generated by adding small additional steps to a PRBS signal. This is because if the same approach is used, the complex nonlinear dynamics of the WWTP lead to an ill-conditioned dataset that can not be used for model identification. By making small changes to a working PRBS input signal, the corresponding output responses would be similar to those obtained with the PRBS signal but are able to capture the system nonlinearity. Here the small steps are added by another PRBS signal with magnitudes $[0, 0.1 \times (\bar{u}_{data} - \underline{u}_{data})]$, where $\bar{u}_{data}$ and $\underline{u}_{data}$ are the upper and lower bounds of the PRBS signal.

System orders of the approximation models are again determined by guess and check such that fair prediction performances are achieved. A fully-connected NN model in the following form is generated:

$$f : \mathbb{R}^{878} \to \mathbb{R}^{410}$$

where $878 = 20 \times 43 + 9 \times 2$, $410 = 10 \times 41$, 43 is the number of inputs plus outputs (2 inputs and 41 outputs), 20 is the number of past data points, 9 is the number of future inputs, and 10 is the number of output steps been predicted. Again the NN is designed to fit the controller design such that the model needs to be called only once while defining the optimization problem.

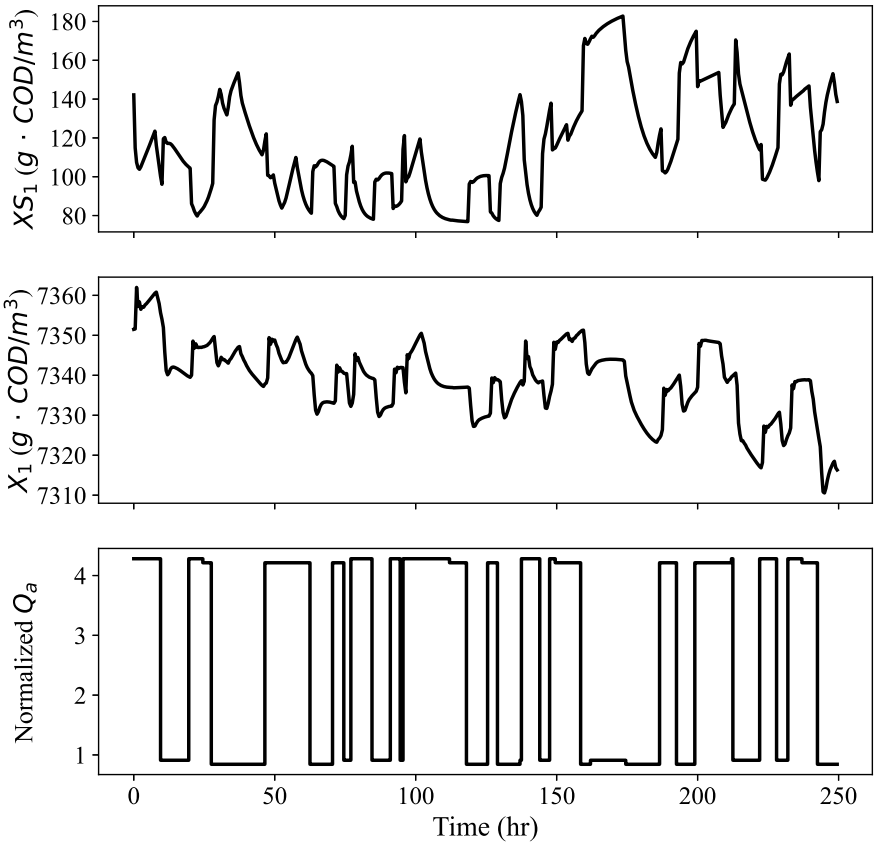

Figure 2.11: Multi-level input and the corresponding output response.

2.4.4 Results and discussion

2.4.4.1 Approximation models

Six approximation models are identified for the WWTP process. Based on the original model, three sets of linearization points are investigated, namely 1, 4, and 5 linearization points. Note that due to the complexity of multi-point linearized TPWL models, simulations are not performed with TPWL models with 145 states. Instead, only TPWL-POD models with 47 artificial reduced states are employed in MPC and EMPC simulations. An LTI model with state vector dimension $s = 48$ is identified using the subspace method. The

Table 2.4: Open-loop response of the reduced model and the original model - WWTP

Model applied	Order	RMSE
1 p.t. linearized	145	0.056
1 p.t. linearized & POD	47	0.057
4 p.t. linearized & POD	47	0.059
5 p.t. linearized & POD	47	0.057
Subspace LTI	48	0.070
Neural Network	N/A	0.067

fully-connected NN with input-output design mentioned in the previous section consists of two hidden layers, of which each layer contains 40 nodes.

Figure 2.12 presents the open-loop response of XS_1 and the corresponding input signal of KL_{a5}. The average NRMSEs of the outputs predicted by approximation models are summarized in Table 2.4. The application of the POD method has little effect on the open-loop prediction performance of the model with a single linearization point. As for the TPWL-POD models, the model performance improves slightly with increased linearization points. The subspace model has the worst performance due to significant information loss caused by limiting to a linear model. The NN model performs slightly better than the subspace model as it captures the nonlinearity of the system to some extent. From Table 2.4, it can be concluded that the single-point-linearized model has the best prediction performance. To summarize, all approximation models respond to input changes without delay.

2.4.4.2 Conventional tracking model predictive control

MPC simulations based on the approximation models introduced in the previous section are carried out and the results are summarized in this section. Table 2.5 presents the average computational costs and the accumulated objective function values of the original model

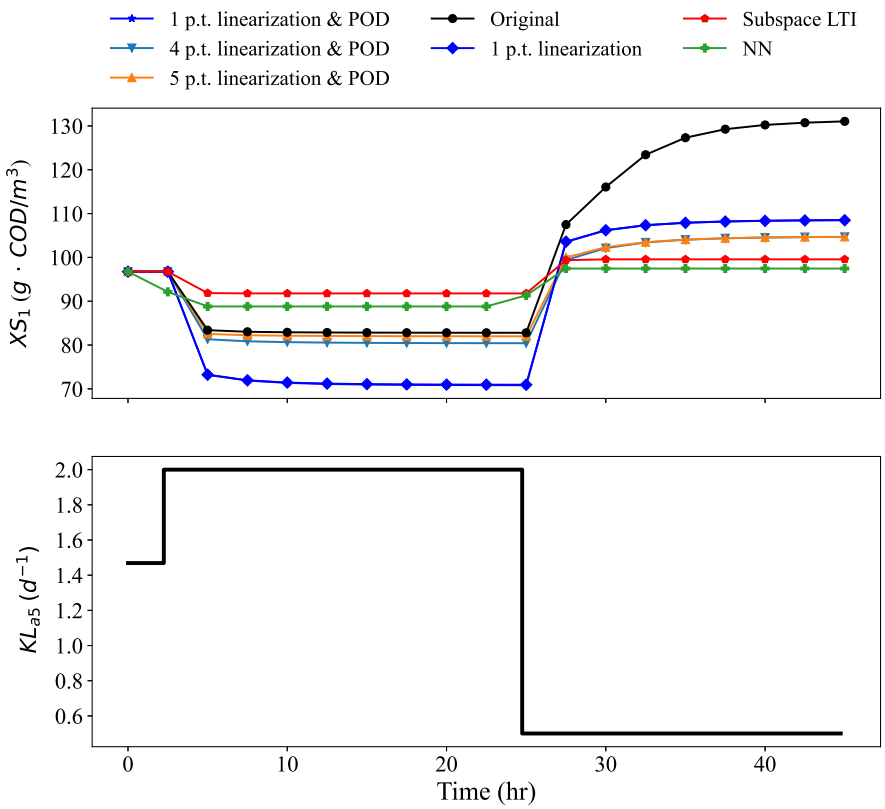

Figure 2.12: Open-loop output responses of the WWTP first-principle model and approximation models.

and the approximation models. The dynamic trajectories of XS_1 under each of the MPC controllers are shown in Figure 2.13, while Figure 2.14 presents the optimal control strategies.

As can be observed from Table 2.5, the single-point linearized model with reduced order converges to the set-point fastest. According to Figures 2.13 and 2.14, it also gives the optimal control strategy closest to that obtained with the original nonlinear model. Note that the model leads to the largest computational cost among first-principle-model-based approximation models, indicating the effect of the number of states considered in the optimization problem. The second fastest convergence to the set-point is given by the single-point linearized model with order reduction. A milder response in the output is also observed from Figure 2.13. The multi-point-linearized TPWL-POD models give similar responses, with more significant overshoots in the output at the beginning. It is also noticed that for in-

Table 2.5: MPC results of the reduced model and the original model - WWTP

Model applied	Order	Time	Objective
Original	145	39.63	1170.32
1 p.t. linearized	145	4.75	3891.11
1 p.t. linearized & POD	47	2.46	1403.76
4 p.t. linearized & POD	47	2.78	4736.07
5 p.t. linearized & POD	47	2.84	4736.07
Subspace LTI	48	0.48	11696.64
Neural Network	N/A	9.62	6647.17

put Q_a, the control strategies obtained based on these two models start further from the set-point. It is also observed that with an increased number of linearization points, the computational complexity increases.

Being the most efficient in terms of computational cost, the subspace model has the worst performance. A huge overshoot is observed at the beginning and it failed to converge to the set-point without offset. As shown in Figure 2.13, the subspace model leads to an underdamped response after the first overshooting peak. These indicate a more significant plant-model-mismatch between the subspace model and the real system. The NN model is the most computationally expensive out of all approximation models considered. This could be due to the less smooth multi-step-ahead prediction trajectory, leading to higher costs in computing Jacobian and Hessian while solving the optimization problem. It shows better performance compared to the subspace model, however offset is as well observed. Figure 2.14 indicates the NN model leads to an significant offset for input Q_a. To be noteworthy, the NN model is the only model that leads to an output response without overshoot in the beginning.

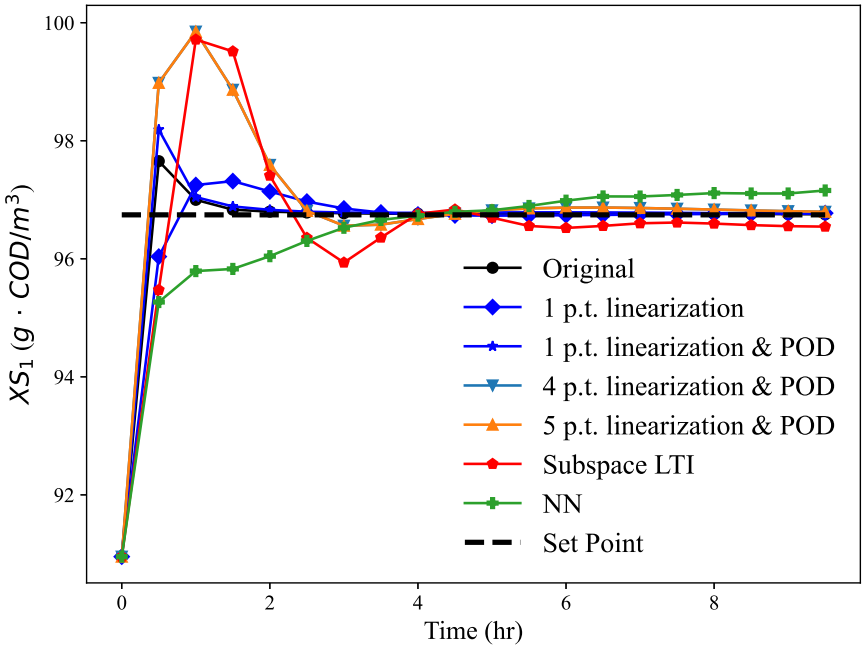

Figure 2.13: Optimal output trajectory of the WWTP under different model-approximation-based conventional MPC frameworks.

2.4.4.3 Economic model predictive control

EMPC simulations are as well performed with the approximation models and the original nonlinear model. Table 2.6 presents the EMPC simulation results of the WWTP process. Figure 2.15 shows the trajectory of the effluent quality (EQ) index under the optimal control strategies obtained with different methods, which is presented in Figure 2.16.

For the first-principle-model-based model reduction, it is observed from Table 2.6 that the single-point linearized model has the best economic performance, which is the same as that observed in MPC simulations. With model order reduction, the computational speed was further improved however with a worse performance at the same time. The time required to solve the optimization problem increases steeply when the number of linearization points increases. The effect of model complexity is noticed to have a greater impact on EMPC compare to MPC. The economic performance however, did not improve when the number

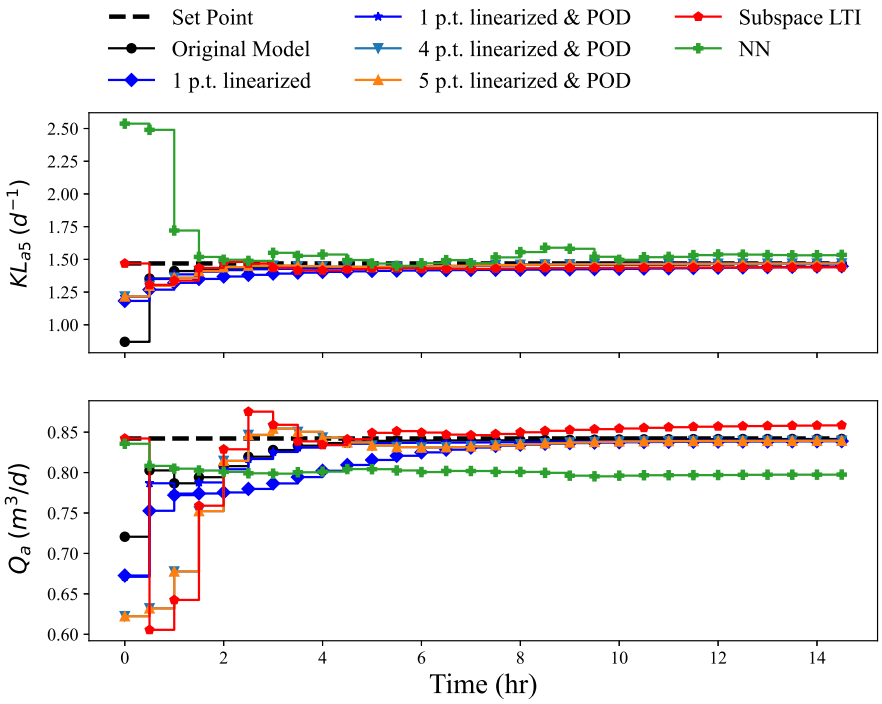

Figure 2.14: Optimal control policy of the WWTP under different model-approximation-based conventional MPC frameworks.

of linearization points increases. This could be because the system operates closely around the steady-state. It can be seen from Figure 2.15 that apart from that obtained with the original model, the best effluent quality is achieved with the single-point TPWL model. The EQ index trajectories of the multi-point TPWL-POD methods are worse and similar to each other, which matches with the results as shown in Table 2.6.

The subspace model leads to EMPC performances better than the multi-point linearized models but worse than the single-point linearized model. The reduction in computational cost is more significant than that achieved with the single-point linearized model as the order of the subspace models is much lower.

The NN-based EMPC framework provides economic performance slightly better than that obtained by the multi-point TPWL-POD models. The computational cost is lower than multi-point TPWL-POD models but higher than the LTI models. This is reasonable as NN is nonlinear by its nature. It is noticed from Figure 2.16 that the optimal control

Table 2.6: EMPC results of the reduced model and the original model - WWTP

Model applied	Order	Time	Objective
Original	145	396.07	115862.71
1 p.t. linearized	145	7.06	118614.70
1 p.t. linearized & POD	47	5.64	122521.16
4 p.t. linearized & POD	47	23.68	138520.05
5 p.t. linearized & POD	47	42.16	138520.05
Subspace LTI	48	6.86	121468.84
Neural Network	N/A	12.65	135274.59

trajectory of Q_a obtained based on the NN osculates the most out of all models considered. This may be caused by the non-smooth dynamics of the NN, leading to challenges while solving the optimization problem.

2.5 Summary

This chapter evaluates the applicability and performance of representative model approximation methods in MPC and EMPC. While these model approximation methods have been successful in open-loop dynamic prediction, the nature of advanced control indeed poses great challenges for these existing model approximation methods as demonstrated in the simulations of this chapter. We have the following remarks:

- Approximation models do not always lead to faster computation. The time required to do one iteration while solving the optimization problem might be reduced by using an approximation model. However, depending on the model structure, the number of iterations required might increase, resulting in a longer computational time.

- The same approximation method could perform very differently on different systems.

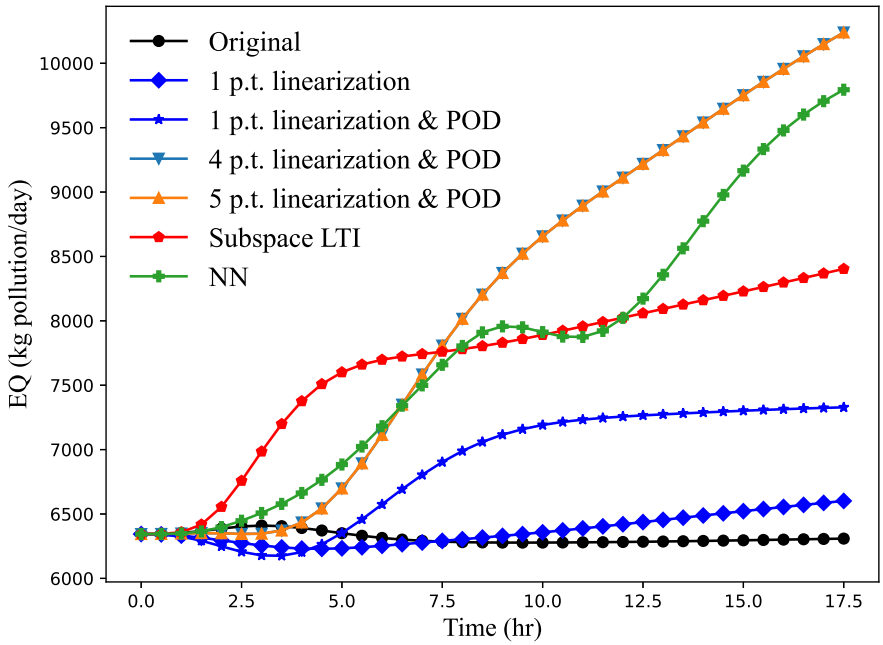

Figure 2.15: Effluent quality (EQ) index trajectory under different model-approximation-based EMPC frameworks.

For example, increased linearization points improve model performance for the alkylation process but not the WWTP process.

- For MPC, model accuracy and structure are essential for achieving satisfying set-point-tracking without offsets. More investigations are necessary for dataset-based approximation approaches.

- EMPC tries to track the trajectory that maximizes the economic performance. This requires the model to have sufficient accuracy over the entire operating range, which poses challenges for the existing methods.

- The computational complexity of POD combined with linearization increases quickly with the increase of the number of linearization points. At the same time, the increased number of linearization points does not necessarily lead to improved control performance in the examples. How to determine the optimal number of linearization

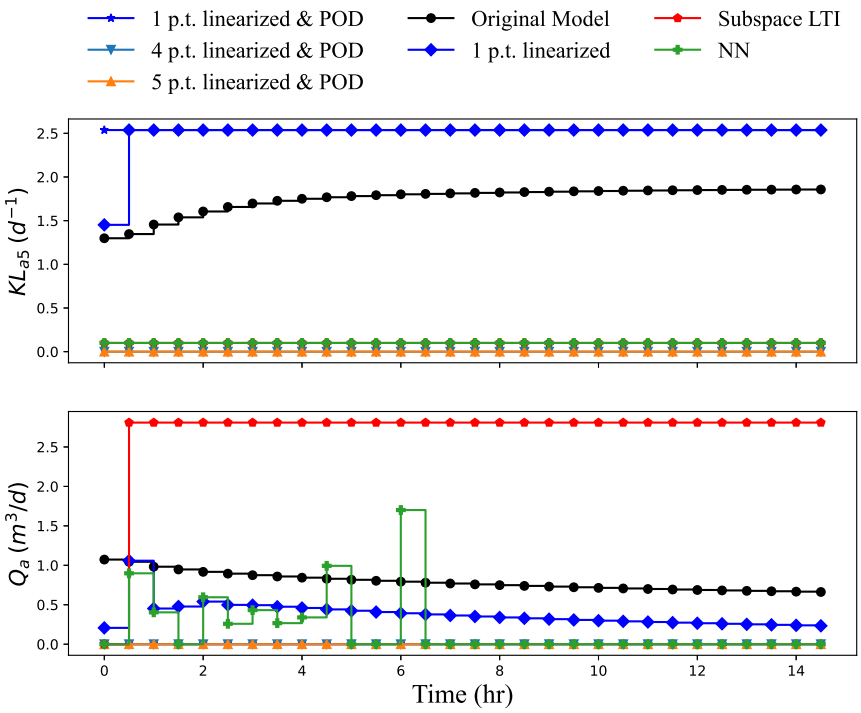

Figure 2.16: Optimal control trajectories under different model-approximation-based EMPC frameworks

points may be a question worth further investigation.

- The NN models identified in this chapter focus on multi-step-ahead prediction. These models may give sufficiently accurate approximations of nonlinear systems. However, oftentimes these models return non-smooth prediction trajectories, which may pose challenges in the associated optimization. More work needs to be done to improve the overall framework. For example, an optimal operating point can be determined with NN-based EMPC first, followed by designing a more accurate controller based on the obtained set-point for better control strategies.

Chapter 3

Model predictive control of agro-hydrological systems based on a two-layer neural network modeling framework

Water scarcity is a rapidly escalating global issue due to various factors such as population growth and climate change. Approximately 70% of the freshwater is consumed by agriculture activities [53, 54]. Thus, improving the water-use efficiency in the agriculture sector is essential. Currently, practical irrigation policies are mostly open-loop, which are determined based on heuristic or empirical knowledge. Real-time feedback from the field is often not considered. This approach typically has low irrigation efficiency, as the applied irrigation amount can be imprecise, leading to over or insufficient irrigation[2].

Increasing attention has been drawn to closed-loop irrigation over the past decade [54, 55]. Among various approaches, advanced control strategies are popular due to their ability to handle constraints and multiple objectives simultaneously [56, 57, 58, 13, 59, 60]. In

[2]This chapter has been published as a journal paper.

[56, 57, 58], conventional model predictive control (MPC) is employed to optimize the soil-moisture dynamics in real-time in the presence of weather disturbances. Instead of tracking a set-point, Mao et al. proposed an MPC controller that tracks a target zone (ZMPC), which provides more degrees of freedom in control actions [13]. Long-term irrigation scheduling that aims to optimize crop production over a longer time duration is investigated in [59, 60]. To be more specific, Nahar et al. proposed a hierarchical framework including a scheduler and a controller [59]. The scheduler has a larger sampling time and optimizes over the entire crop growth season. Some soil moisture reference is provided by the scheduler and is tracked by the controller over shorter horizons. Sahoo et al. proposed a knowledge-based scheduler that optimizes the irrigation amount and time explicitly and is shown to be effective under different scenarios [60].

One of the major challenges in applying advanced control to the agro-hydrological system is the high computational cost. The water dynamics are often described by the Richards equation, which is a nonlinear PDE developed based on first principles [61]. Spatial discretization is often required to simplify the Richards equation in applications [61], however, the resulting nonlinear ODE system is still difficult to solve and leads to states with higher dimensions.

Model approximation based on neural network (NN) is a rather generalized approach compared to traditional model identification. Motivated by the accuracy and computational efficiency of the NN model reduction observed in Chapter 2 and inspired by the structure of the LPV model developed in [13], we propose a two-layer NN framework to approximate the agro-hydrological dynamics in this chapter. The proposed framework is shown to have better open-loop prediction performance compared to a benchmark single long short term memory (LSTM) NN.

The developed two-layer NN model is employed in a ZMPC controller in the presence of noise and weather disturbances. The control objective is to minimize the amount of irrigation required while keeping the soil moisture content inside the target zone. In order to

address the plant-model-mismatch, open-loop bias correction is added to the model, while a shrinking target zone is employed for ZMPC. Two bias correction models are investigated with different updating frequencies and the one with better open-loop prediction performance is used in ZMPC. The shrinking ZMPC target zone is designed such that the shape of the zone can be tuned by modifying the hyper-parameters, of which the effects on the control performance are investigated.

This chapter is organized as follows. The agro-hydrological system of interest is introduced in Section 3.1. Details of the proposed two-layer NN framework are discussed in Section 3.2, of which the open-loop validation is provided in Section 3.3 with two plant-model-mismatch compensation strategies investigated in Section 3.4. The ZMPC controller employed is introduced in Section 3.5. The closed-loop simulation results are presented in Section 3.6. Finally, concluding remarks are provided in Section 3.7.

3.1 Preliminaries

A schematic of the agro-hydrological system of interest is shown in Figure 3.1 with grass being the crop of interest. The soil properties are assumed to be homogeneous horizontally and only the soil water dynamics on the vertical axis are considered. Irrigation is applied with a sprinkler and is assumed to reach the soil surface evenly. Precipitation and evapotranspiration are considered as known disturbances based on weather forecasting data. It is assumed that precipitation reaches the soil surface in identical ways as irrigation.

3.1.1 Soil Water Dynamics

Based on first principles, the soil water dynamics can be modeled using the 1-dimensional Richards equation with the crop and weather information taken into account [61]:

$$c(h)\frac{\partial h}{\partial t} = \frac{\partial}{\partial z}\left[K(h)\left(\frac{\partial h}{\partial z}+1\right)\right] - \alpha(h)\frac{K_c ET_0}{z_r} \tag{3.1}$$

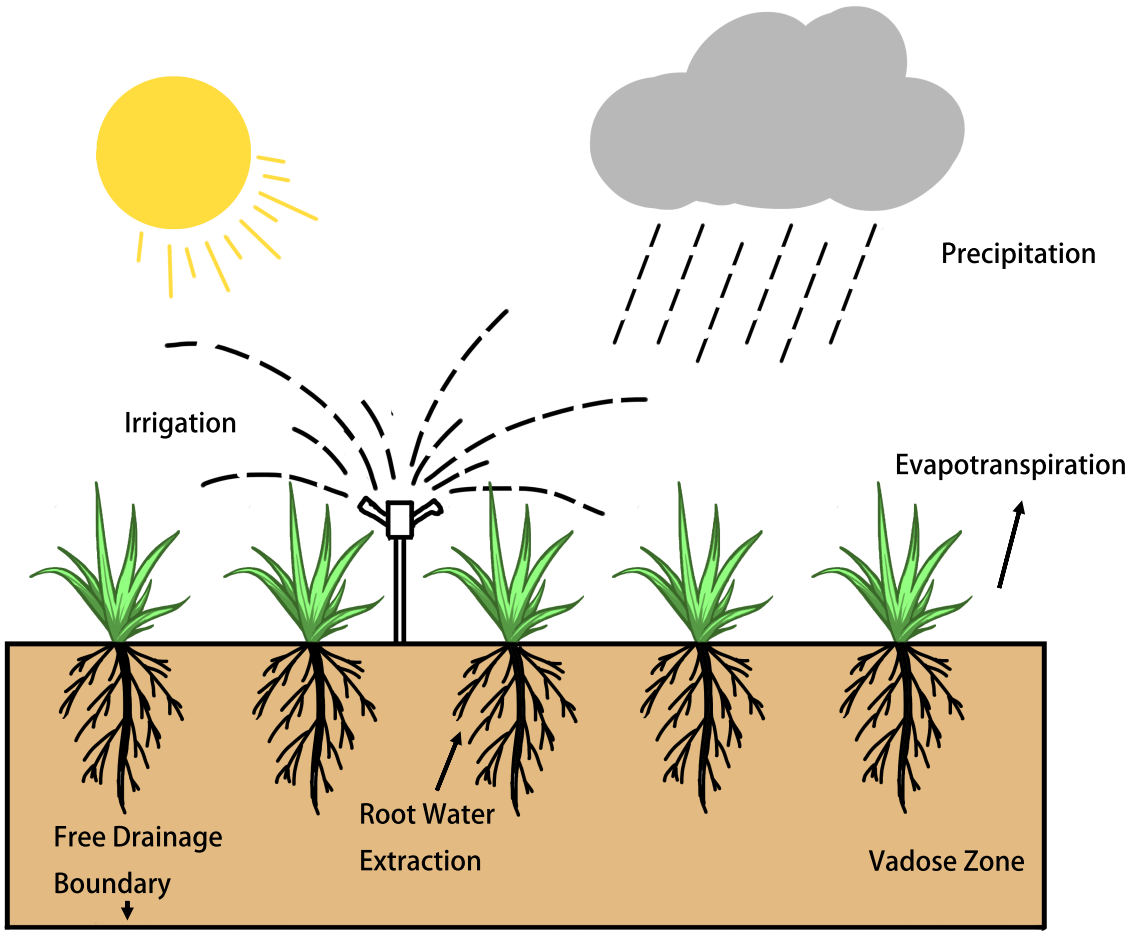

Figure 3.1: A schematic diagram of the considered agro-hydrological system.

where h $[m]$ is the capillary potential, c $[m^{-1}]$ denotes the soil capillary capacity, and K $[m/s]$ is the soil hydraulic conductivity. The $-\alpha(h)\frac{K_c ET_0}{z_r}$ term captures the effect of weather and crop, where $\alpha(h)$ and K_c are dimensionless factors, namely the water stress factor and the crop coefficient. ET_0 $[m]$ denotes the reference evapotranspiration and is assumed to be a known weather-dependent disturbance. $z_r = -0.13$ m represents the rooting depth of the crop and is assumed to be time-invariant in this chapter. Note that z $[m]$ represents the vertical axis of the soil with upwards as the positive direction. $c(h)$ and $K(h)$ are modeled by the van Genuchten-Mualem soil hydraulic model [62]:

$$c(h) = (\theta_s - \theta_r)\alpha n\left(1 - \frac{1}{n}\right)(-\alpha h)^{n-1}[1 + (-\alpha h)^n]^{\frac{1}{n}-2} \tag{3.2}$$

Table 3.1: Soil parameters of sandy loamy soil

$K_s\,[m/s]$	1.23e-5
$\theta_s\,[m^3/m^3]$	0.41
$\theta_r\,[m^3/m^3]$	0.065
$\alpha\,[m^{-1}]$	7.5
n	1.89
m	0.47

$$K(h) = K_s\Big[(1+(-\alpha h)^n)^{-\left(1-\frac{1}{n}\right)}\Big]^{\frac{1}{2}}\left[1-\left[1-\left[(1+(-\alpha h)^n)^{-\left(1-\frac{1}{n}\right)}\right]^{\frac{n}{n-1}}\right]^{1-\frac{1}{n}}\right]^2 \tag{3.3}$$

where K_s $[m/s]$ denotes the saturated hydraulic conductivity, θ_s $[m^3/m^3]$ and θ_r $[m^3/m^3]$ represent the saturated and the residual soil moisture, respectively. α $[m^{-1}]$ and n are the van Genuchten-Mualem parameters related to the soil properties. Sandy loam soil is considered in this chapter and the corresponding parameters are adopted from [63] and are presented in Table 3.1.

3.1.2 Boundary Conditions

The boundary conditions employed in this chapter are presented in (3.4)–(3.5). The Neumann boundary condition is used at the soil surface as shown in (3.4), while free drainage is assumed for the bottom boundary as described in (3.5).

$$\left.\frac{\partial h(t)}{\partial z}\right|_T = -1 - \frac{I(t)+P(t)}{K(h(t))} \tag{3.4}$$

$$\left.\frac{\partial h(t)}{\partial z}\right|_B = 0 \tag{3.5}$$

where $I(t)$ $[m/s]$ and $P(t)$ $[m/s]$ denote the rate of irrigation and precipitation at time t. Note that interception caused by grass leaves is ignored in this chapter, only evapotranspiration, precipitation, and irrigation are considered.

3.1.3 Model Discretization

The system of equation (3.1) is discretized over the vertical spatial axis following the approach proposed in [61], which leads to an ODE system:

$$\dot{x}(t) = f(x(t), u(t)) \tag{3.6}$$

where $x \in \mathbb{R}^{26}$ and $u \in \mathbb{R}^1$ are the state and input vectors respectively, while f defines the nonlinear system dynamics. The total depth of soil considered is 0.5 m and is evenly discretized into 26 nodes. The water dynamics at the center of each node are used to represent the dynamics of the entire node. In this chapter, irrigation rate $I(t)$ $[m/s]$ is the system input, and h $[m]$ at the discretized nodes represents the system states. The system output $y \in \mathbb{R}^1$ is the volumetric water content θ $[m^3/m^3]$ at the rooting depth z_r, which is an algebraic function of the state vector defined by the soil-water retention equation of van Genuchten (3.7). Eqn. (3.6) and Eqn. (3.7) are used to generate all training datasets.

$$\theta(h) = (\theta_s - \theta_r)\left[\frac{1}{1 + (-\alpha h)^n}\right]^{1-\frac{1}{n}} + \theta_r \tag{3.7}$$

3.2 Proposed Two-Layer Neural Network

To reduce the computational complexity of the first-principle model, a two-layer NN framework is proposed to approximate its nonlinear dynamics. In this section, details regarding the proposed framework will be discussed. Section 3.2.1 introduces the motivation and provides an overview of the proposed framework. Sections 3.2.2 and 3.2.3 discuss the data generation procedure for the individual layers of the framework respectively.

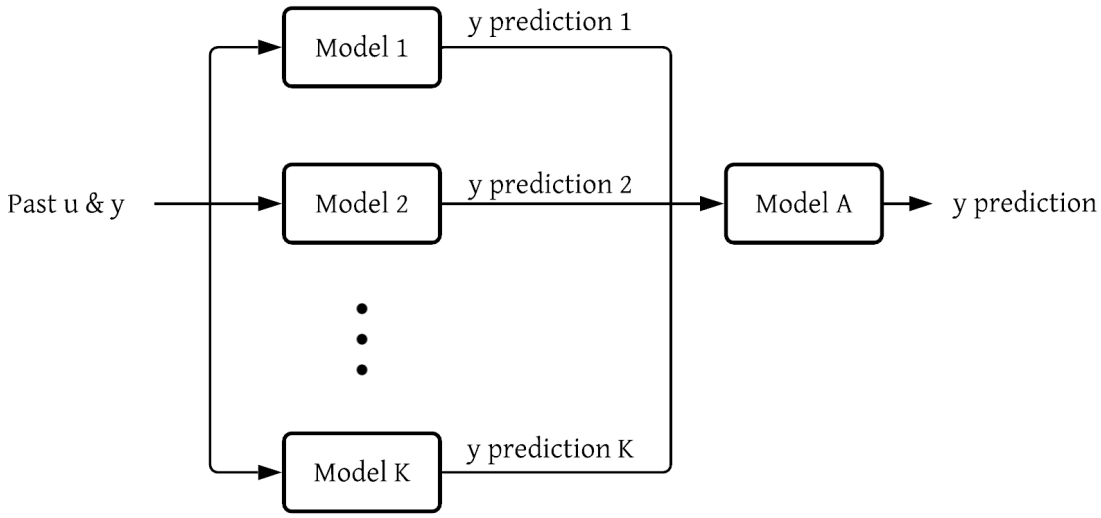

Figure 3.2: A schematic diagram of the proposed two-layer NN framework.

3.2.1 Overview

In this chapter, LSTM NNs will be used to approximate the original nonlinear system dynamics due to their ability of efficiently handle sequential data [64, 65]. The motivation for designing the proposed framework is that a single LSTM is not able to capture the complex nonlinear dynamics of the agro-hydrological system over the entire operating range of interest. It was noticed, however, within a smaller operating region, a single LSTM provides satisfying modeling performance. Detailed simulation results will be presented in Section 3.3.

A schematic diagram of the proposed framework is presented in Figure 3.2. The first layer contains K sub-models, namely Model 1– Model K, which will be referred to as $M_1, M_2, \cdots, M_K$ in the following text. The second layer consists of a single model (Model A, referred to as M_A). Each sub-model in the first layer is responsible for a sub-operating region. The union set of the sub-operating regions forms the entire operating region of interest. $M_1 - M_K$ are LSTMs while M_A is a fully connected NN. At the current time step t, $M_1 - M_K$ take the system input u and output y obtained from p previous time steps as NN

Table 3.2: Operating ranges of the sub-models

	Operating range	# of LSTM layers
M_1	$y \in [0.12, 0.27]$	1
M_2	$y \in [0.21, 0.32]$	1
M_3	$y \in [0.29, 0.40]$	2

inputs:

$$NN_{in}(t) = \begin{bmatrix} u(t-p+1), y(t-p+1) \\ u(t-p+2), y(t-p+2) \\ \cdots \\ u(t), y(t) \end{bmatrix},$$

and returns K one-step-ahead predictions regarding the system output:

$$NN_{out_1} = [\bar{y}_1(t+1), \cdots, \bar{y}_K(t+1)]^T$$

Recall that each sub-model is only reliable in a sub-operating region. Inspired by Mao et al. [13], the proposed framework adopts the idea of the LPV models. The model performance can be improved by combining predictions made by multiple sub-models. The one-step-ahead predictions made by $M_1 - M_K$ are fed to M_A, where the final prediction of the system output at the next time step ($NN_{out_2} = y_{pred}(t+1)$) is obtained.

Overall, at any time step t, the proposed framework takes $NN_{in}(t)$ as input and returns the one-step-ahead prediction of the system output $y(t+1)$, which can be expressed as a nonlinear function:

$$\hat{y}(t+1|t) = g(NN_{in}(t)) \tag{3.8}$$

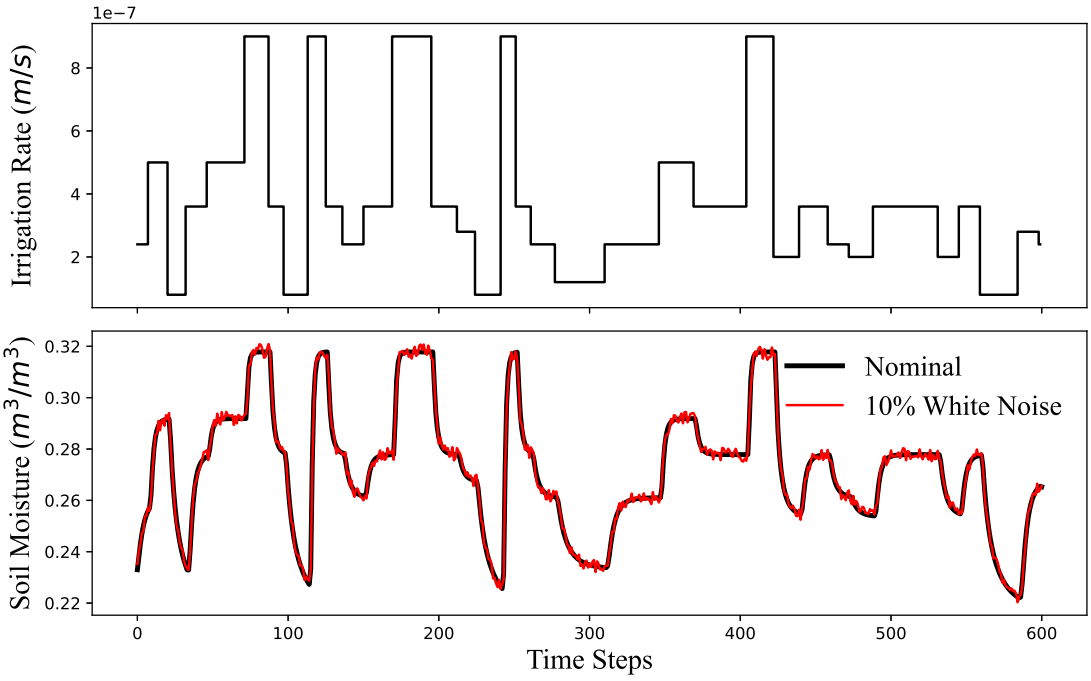

Figure 3.3: A segment of the training dataset of M_2

3.2.2 Design and training of first-layer NNs

As discussed in the previous section, the first layer of the proposed framework consists of K LSTMs. In this chapter, $K = 3$ is used. The operating regions in terms of the system output for each sub-model are summarized in Table 3.2. The number of LSTM layers employed in the models is also included. Each model consists of 1 or 2 LSTM layers, followed by a fully connected layer. Activation functions `'sigmoid'` and `'tanh'` are used. Both activation functions have smooth dynamics, where `'tanh'` function provides steeper responses compare to `'sigmoid'` function. Each sub-model is equivalent to a nonlinear function:

$$m_i : \mathbb{R}^{20\times 2} \to \mathbb{R}^1, \quad i = 1, 2, 3$$

where the model input is a matrix with 20 rows and 2 columns, and the output is a scalar. 20 is the number of the past input-output data points ($p = 20$), while 2 is the number of system input ($n = 1$) plus the number of system output($l = 1$). All models have a sampling

time of 2 hours.

Three training datasets that vary in the corresponding operating region are generated for the sub-models respectively. Each dataset contains 30000 data points. Multi-level pseudorandom input signals are employed to generate all three datasets, which are fed into the discrete ODE system (3.6) and the algebraic equation (3.7) for open-loop simulations. A multi-level pseudorandom signal is similar to a pseudorandom binary signal, except that it has multiple levels, which helps to stimulate the nonlinear dynamics of the agro-hydrological system. More details regarding the design of multi-level pseudorandom inputs can be found in our previous work [66]. Figure 3.3 presents a segment of the training input signal and the corresponding system output for M_2. Note that a random noise signal ϵ is added to the output trajectory to mimic real process operations, which is shown in Figure 3.3(b) in red. The max magnitude of ϵ is defined as follows:

$$\epsilon_{max} = 0.1[y_{max} - y_{min}]$$

where y_{max} and y_{min} are the maximum and minimum value of the system output in the dataset.

All datasets need to be reorganized to match the input-output structure of the NNs. Scaling is also required such that the magnitudes of the scaled data are less than 1. All simulations are carried out in Python. TensorFlow [67] is used to train the NN models. Readers may refer to our previous work [66] for more details regarding data pre-processing and NN training.

3.2.3 Design and training of second-layer NN

The second layer of the proposed framework consists of a fully connected NN equivalent to the following nonlinear function:

$$m_A : \mathbb{R}^3 \to \mathbb{R}^1$$

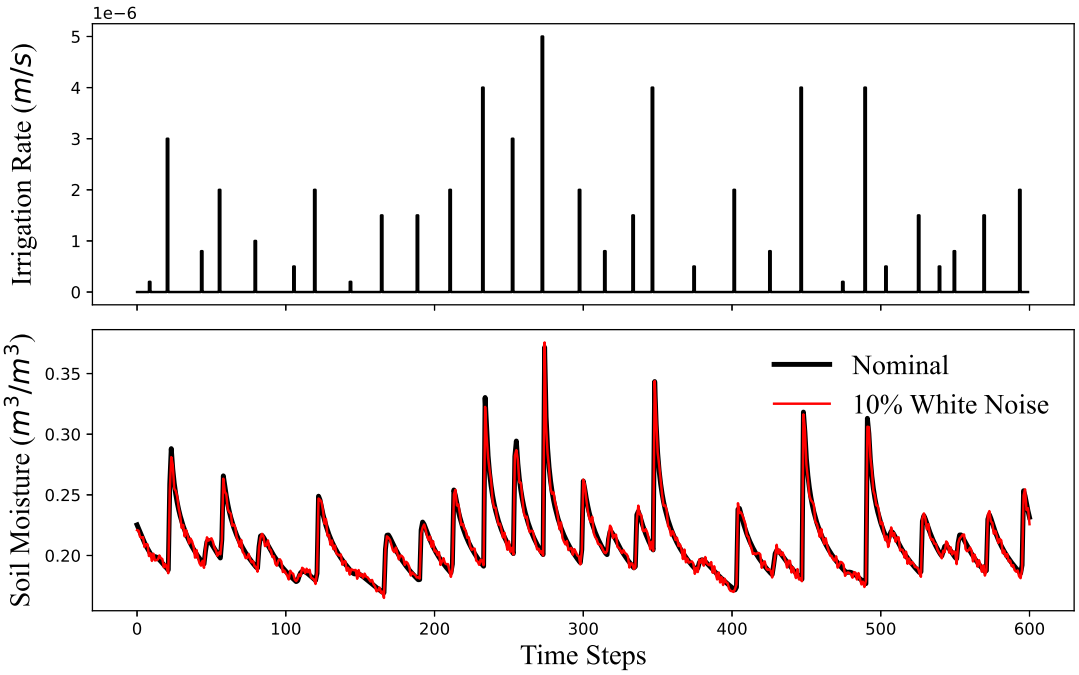

Figure 3.4: A segment of the training dataset of M_A

The inputs to model M_A are the one-step-ahead predictions of the system output made by M_1, M_2 and M_3, while the output is the actual system output prediction. Two fully connected layers with the activation function `'sigmoid'` are used in M_A.

Under the objective of predicting the system output over the entire operating range of interest, the training dataset of M_A covers the entire operating range. A section of the training dataset including 100000 points is presented in Figure 3.4, where the input signal and the corresponding output trajectories without and with white noise added are presented. Note that the training dataset of M_A spans all the sub-operating regions with more aggressive dynamics. Instead of continuous irrigation with lower magnitudes, an impulse irrigation signal is used to provide a more aggressive output response that varies in a larger operating range, which is desired for M_A training.

Similar to that discussed in the previous section, reconstruction and scaling are necessary for this dataset. Note that the inputs to M_A are not the system input and output. The dataset is passed through the pre-identified sub-models after the data pre-processing. The

Table 3.3: NRMSE of multi-step-ahead-predictions of the sub-models

# of prediction steps	M_1	M_2	M_3
1	0.015	0.016	0.073
10	0.042	0.049	0.111
20	0.060	0.050	0.111

predictions made by the sub-models are collected, which are the designed inputs for M_A and are used to train M_A.

3.3 Model Validation

The prediction performance of the proposed framework is validated in this section. The maximum number of prediction steps is $N = 20$ for this chapter, which is also the control horizon of interest for following ZMPC applications. Section 3.3.1 presents the validation results of the sub-models. The proposed framework is then compared with a single LSTM in Section 3.3.2.

3.3.1 Validation of the first-layer sub-models

The multi-step-ahead prediction performances of $M_1 - M_3$ are presented in this section. Validation datasets are obtained in similar manners as the training datasets with 20% white noise added to the system output. This helps to better test the prediction performance of the models. At any given time step, noise-treated validation data are fed to the model for one-step-ahead prediction. Starting from the second time instant, the prediction made by the model at the previous time instant will be used as the initial condition of the current time instant. This procedure is repeated 20 times, providing twenty-steps-ahead predictions.

The prediction performance is measured by the normalized root mean square error (NRMSE) of the multi-step-ahead predictions in the desired operating range of each model

and is summarized in Table 3.3. The NRMSE is defined as follows:

$$NRMSE = (y_{max} - y_{min})^{-1} \sqrt{\frac{\sum_{i=1}^{t_f} (y_{act}(i) - y_{pred}(i))^2}{t_f}} \tag{3.9}$$

where t_f is the total number of validation data points, y_{act} and y_{pred} are the true and predicted system outputs, respectively. Figures 3.5, 3.6, and 3.7 present the multi-step-ahead open-loop prediction performance of M_1, M_2, and M_3 respectively. The same line styles are used in Figures 3.5–3.7 for consistency.

All three models provide reasonable multi-step-ahead prediction without delays in the presence of noise. When the magnitude of the system output is low, small prediction offsets can be observed for M_1 in Figure 3.5. Figure 3.6 shows only some minor offsets exist at the local extremes for M_2. For both M_1 and M_2, the effect of noise seems to be minor. For M_3, Figure 3.7 indicates that the noise has a stronger impact on the single-step-ahead prediction, which deviates and becomes insignificant as the number of prediction steps increases. It is also noticed that when the output converges to steady-states, the impact of the noise is the most significant. Similar to M_1 and M_2, M_3 also has some minor offsets at the local extremes.

For different operating ranges, different system output dynamics are observed. When the soil moisture content and the irrigation rate are low, no obvious steady-state is observed (Figure 3.5). As the operating range shifts to the higher magnitude side, the soil moisture content tends to saturate to steady-states under continuous irrigation (Figure 3.7).

3.3.2 Validation of the proposed two-layer framework

The multi-step-ahead prediction performance of the proposed framework is investigated in this section. A single LSTM that aims to predict the agro-hydrological dynamics over the entire operating region is used as the benchmark approximation model. The benchmark LSTM has the same input (NN_{in}) and output (NN_{out_2}) structures as the proposed two-layer

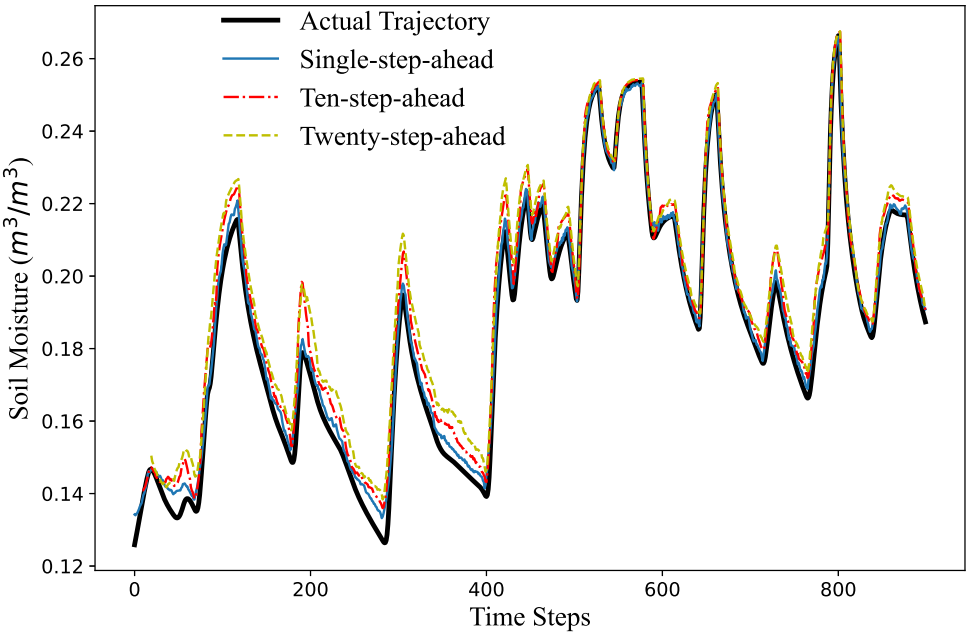

Figure 3.5: One-step-ahead prediction (solid), ten-step-ahead prediction (dash-dotted), and twenty-step-ahead prediction (dashed) of M_1 compared to the actual trajectory (dark solid).

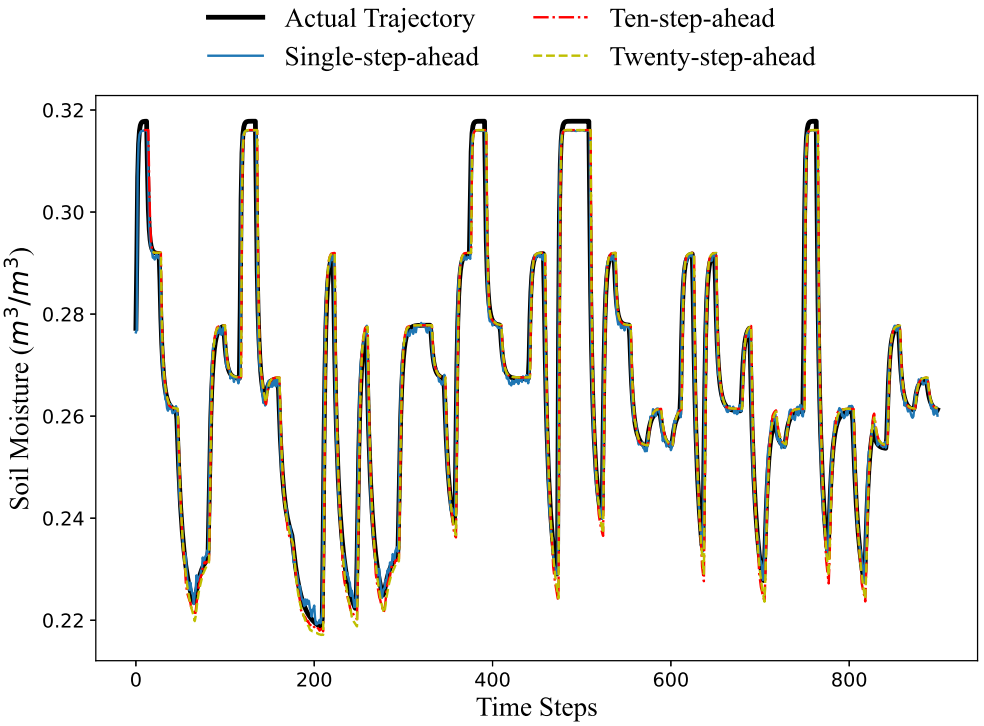

Figure 3.6: Multi-step-ahead prediction performance of M_2.

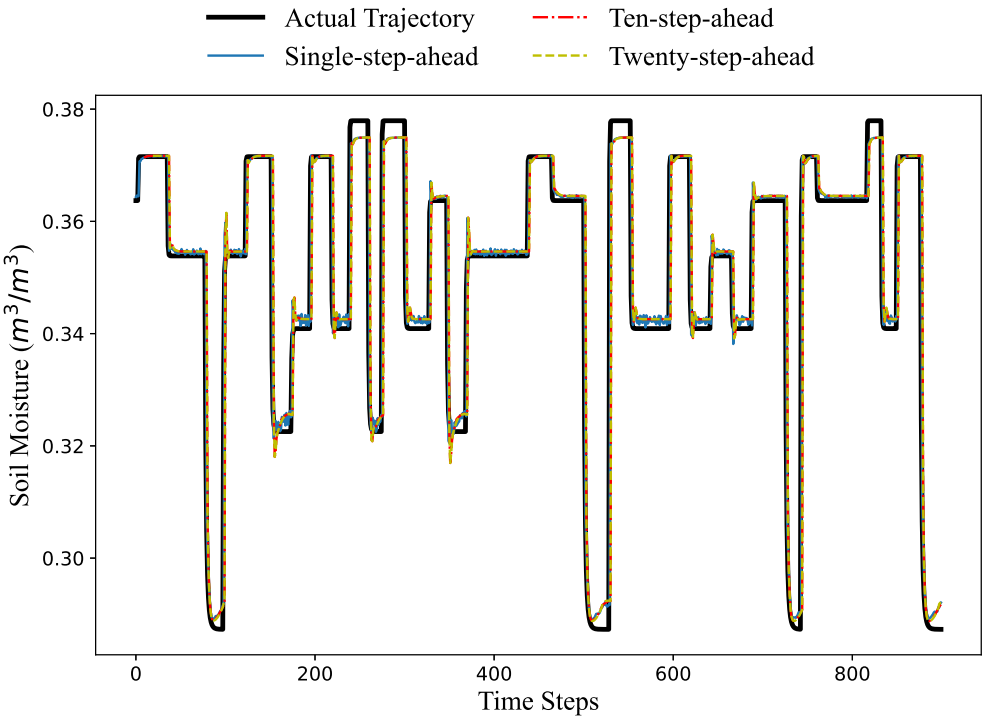

Figure 3.7: Multi-step-ahead prediction performance of M_3.

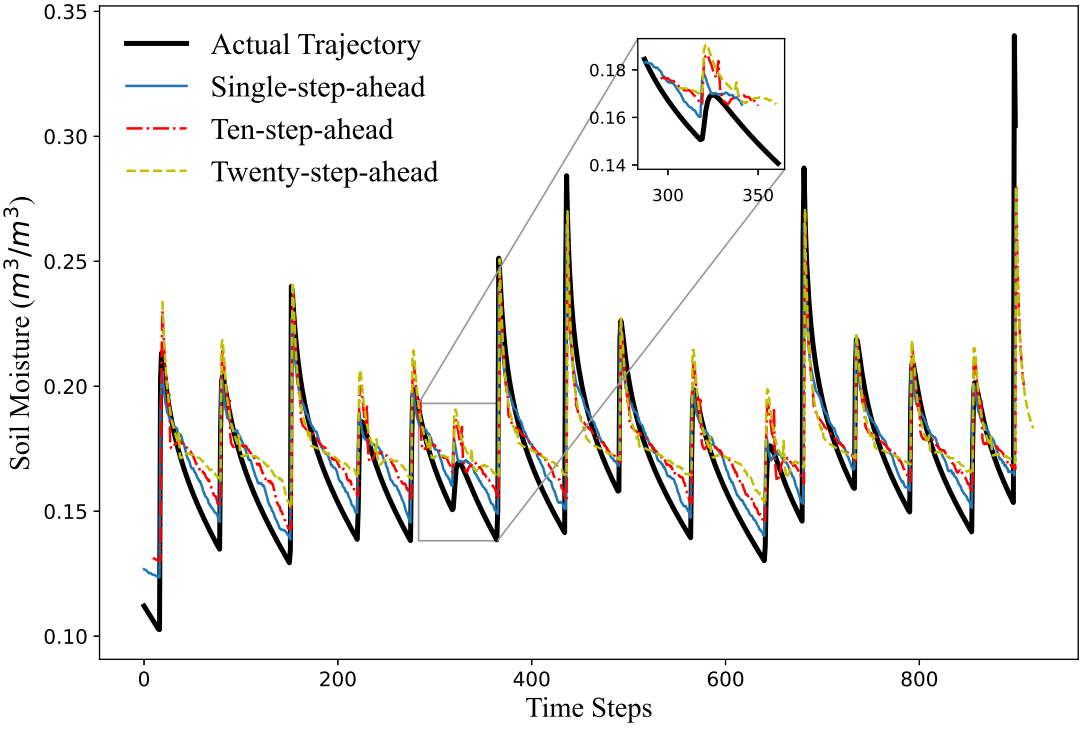

Figure 3.8: Multi-step-ahead prediction performance of an single LSTM.

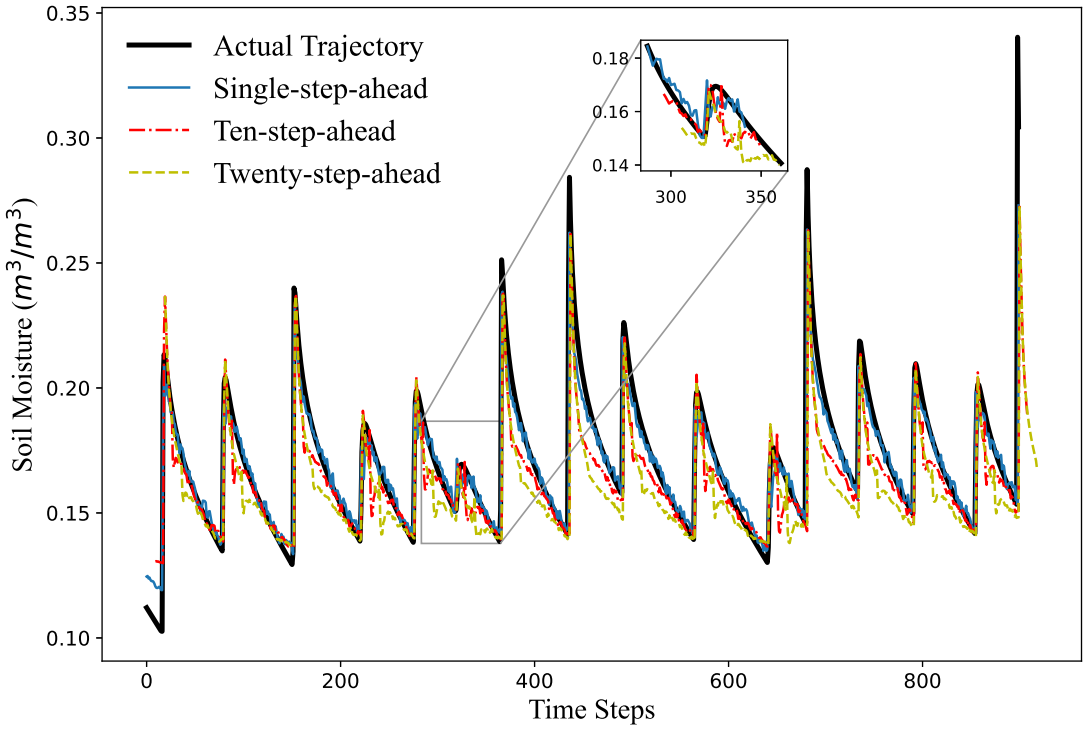

Figure 3.9: Multi-step-ahead prediction performance of the two-layer NN framework.

NN framework, which are discussed in Section 3.2.1. The single LSTM has 2 LSTM layers followed by a fully connected output layer. Both the LSTM layers use the activation function `'sigmoid'` while the output layer uses the activation function `'tanh'`. Note that the dataset employed for training the benchmark LSTM is the same as that used to train the two-layer NN model.

The same validation dataset is used to compare the performance of the two approximation approaches, which spans the entire operating range of interest. The open-loop prediction trajectories of the system output based on the benchmark LSTM and the proposed framework are presented in Figures 3.8 and 3.9 respectively. The single-step-ahead, ten-step-ahead, and twenty-step-ahead predictions are presented in each figure with the same line styles as those employed in the previous section. The NRMSE values under the two cases and the percentage difference $\triangle$ between them are summarized in Table 3.4. The percentage

Table 3.4: NRMSE of multi-step-ahead-predictions of the NN models

# of Pred. steps	Single LSTM	two-layer NN	% Diff. $\triangle$
1	0.076	0.044	42.5
10	0.096	0.082	14.5
20	0.100	0.093	7.14

difference is calculated with respect to the NRMSE of the benchmark LSTM:

$$\triangle = \frac{NRMSE_{LSTM} - NRMSE_{proposed}}{NRMSE_{LSTM}}$$

As presented in Figure 3.8, the LSTM can respond to aggressive dynamics efficiently without delays. However, significant offsets are observed at the local extremes with larger magnitudes. A section of the trajectories is amplified, which can be seen on the top-right corner of Figure 3.8. At both the local maximum and the minimum, significant offsets can be observed in the zoomed-in plot, of which the predicted local minimums are at the level of the true local maximum. Better prediction performance is observed in Figure 3.9. To be specific, the dynamics at the local minimums can be captured by the proposed two-layer framework much better compared to the LSTM. From the zoomed-in window on the top right, it can be observed that the predictions of the local minimum and maximum are more accurate compared to the LSTM. Table 3.4 indicates comparison results, showing that the NRMSE values of the proposed framework are significantly lower than those obtained with the LSTM for all different prediction steps. The improvement of the proposed framework is the most significant with one-step-ahead predictions and slowly reduces as the prediction steps increase.

It is noticed that the proposed framework cannot accurately approximate the local maximums with larger magnitudes. To account for this problem, we introduce a bias-and-scale-updated model to capture the plant-model-mismatch, of which the details are provided in the following section.

3.4 Plant-Model-Mismatch Correction

To capture the plant-model-mismatch, two correction approaches are investigated based on the proposed two-layer NN framework. The first approach is to add a simple bias term b_1 based on the prediction error:

$$y_{act} = y_{pred} + b_1 \tag{3.10}$$

where y_{act} and y_{pred} represent the actual output measurement and the prediction made by the two-layer NN framework respectively.

The second approach is to fit a first order linear model that maps the predicted output to the actual measurement:

$$y_{act} = a \cdot y_{pred} + b_2 \tag{3.11}$$

where a and b_2 are parameters to be identified.

For both approaches, an updating frequency f of the parameters needs to be determined first. In this chapter, we are interested in accurately predicting the system dynamics 20 steps into the future, thus the updating frequency should be a divisor of 20. To test the performance of the two approaches, the validation dataset used in the previous section is employed. The updating policies of the two approaches are presented in Algorithms 1 and 2, respectively. In both algorithms, t represents the current time step, i denotes the step of predictions made ahead of time t, and $\eta(i)$ represents the prediction error at step i.

At any given time instant t, the parameters are first initialized with some initial guess. The initial condition $NN_{in}(t)$ is provided to the proposed framework for multi-step ahead predictions. In the next $N = 20$ steps, predictions will be made with the correction model considered, which will be updated every f time steps for $\frac{N}{f}$ times.

The updating policy of the first approach is to take the average of the accumulated prediction errors over the last f prediction steps. For the second approach, a constrained optimization problem that finds the best-fit parameters for the linear model is solved. Note

that the parameters are held in bounded compact sets, which help to limit the parameters in a reasonable range. To be more specific, $a \in [0.8, 1.5]$ and $b_2 \in [-0.2, 0.3]$ are used. If a simple linear-regression function is employed, the resulting parameters might be impractical. The toolbox CasADi [50] is employed to solve the optimization problems in Python.

Algorithm 1 Parameter updating algorithm for the first correction approach

Set the number of prediction steps $N = 20$
Set the updating frequency $f = f$
Initialize $b_1 = 0$
At each time step of the dataset t, set $i = 1$
Provide the model with the initial condition $y_{pred}(0) = y_{act}(t)$
while $i \leq N$ **do**
  Predict the future system output $y_{pred}(i) = NN_{out_2}(i) + b_1$
  Record the error with respect to the actual output value $\eta(i) \leftarrow (y_{act}(t+i) - y_{pred}(i))$
  if f is a divisor of i **then**
    Update $b_1 = \frac{1}{f} \sum_{j=i-f+1}^{i} \eta(j)$
  end if
  Set $i = i + 1$
end while

Algorithm 2 Parameter updating algorithm for the second correction approach

Ensure: $a \in \mathbb{A}$, $b_2 \in \mathbb{B}$
Set the number of prediction steps $N = 20$
Set the updating frequency $f = f$
Initialize $a = 1$, $b_2 = 0$
At each time step of the dataset t, set $i = 1$
Provide the model with the initial condition $y_{pred}(0) = y_{act}(t)$
while $i \leq N$ **do**
  Predict the future system output $y_{pred}(i) = a \cdot NN_{out_2}(j) + b_2$
  if f is a divisor of i **then**
    Update a and b_2 by solving the optimization problem
    $a^*, b_2^* = \min_{a,b_2} \sum_{j=i-f+1}^{i} (y_{act}(t+j) - y_{pred}(i))$
  end if
end while

Recall the prediction horizon of interest $N = 20$. Thus updating frequencies $f = 1, 2, 5, 10$ are tested and compared. The correction performance is measured by the average absolute prediction error υ_{mm}:

$$\upsilon_{mm} = \frac{1}{N \cdot t_f} \sum^{t_f} \sum^{N} |y_{act} - \hat{y}_{pred}|$$

Table 3.5: Performance of the mismatch correction approach in terms of average absolute error

f	No corr.	Single bias	Linear Model
1	0.101	0.066	-
2	0.101	0.064	0.063
5	0.101	0.195	0.090
10	0.101	0.265	0.098

where the same definitions are kept for all variables. The absolute prediction errors of the two correction approaches are summarized in Table 3.5. The absolute prediction error of just the proposed NN framework is as well included in Table 3.5 and is used as the comparison benchmark.

With $f = 5, 10$, the single-bias correction approach leads to more significant errors, which is not applicable. The linear model correction shows very minor performance improvements. When $f = 2$, similar improvements in the prediction performance are observed in both approaches. The single bias correction performs similarly for $f = 1$ and $f = 2$. For the linear model correction, $f = 1$ is not applicable. Considering the computational complexity of the two approaches, the single-bias correction with $f = 2$ is selected.

3.5 Controller Design

A ZMPC controller with a shrinking target zone is employed in this chapter and is presented in this section. The major target here is to provide sufficient irrigation such that the crop can live and grow healthily, which implies the soil moisture content at the crop root zone needs to be kept in a bounded range instead of one particular point. Compared to a conventional tracking MPC, ZMPC provides more degrees of freedom to the system and can handle economic objectives better, which fits the objective of this chapter.

In the basic design of a ZMPC controller, the target zone is kept time-invariant over the control horizon. In this chapter, we propose a ZMPC design with a time-variant target zone that shrinks over the control horizon to handle plant-model-mismatch. The proposed ZMPC

formulation is presented as follows:

$$\min_{u(t_j|t_i)\in S(\triangle)} \sum_{j=i}^{i+N-1} ||\hat{y}(t_j|t_i) - y_z(t_j|t_i)||_Q^2 + ||u(t_j|t_i)||_R^2 \tag{3.12a}$$

$$\text{s.t.} : \hat{y}(t_{j+1}|t_i) = g(NN_{in}(t_j|t_i)), \quad j = i, i+1, \cdots, i+N-1 \tag{3.12b}$$

$$\hat{y}(t_i|t_i) = y(t_i) \tag{3.12c}$$

$$u(t_j|t_i) \in \mathbb{U}, \quad j = i, i+1, \cdots, i+N-1 \tag{3.12d}$$

$$\hat{y}(t_j|t_i) \in \mathbb{Y}, \quad j = i, i+1, \cdots, i+N-1 \tag{3.12e}$$

$$y_z(t) \in \mathbb{Y}_z(t_j), \quad j = i, i+1, \cdots, i+N-1 \tag{3.12f}$$

where u, y are the system input and output vector, y_z is the zone-tracking slack variable. t_i represents the current time step, $u(t_j|t_i)$ and $\hat{y}(t_j|t_i)$ denote the system input and output at future time t_j predicted at the current time t_i. (3.12a) is the objective function, where Q and R are diagonal weighting matrices for the zone-tracking objective and the irrigation amount respectively. The control objective is to drive the system into a particular operating zone while minimizing the irrigation amount at the same time. (3.12b) represents the two-layer NN framework discussed in the previous section, which is used to predict the system dynamics under the selected input trajectory over the control horizon N. (3.12c) defines the initial state $y(t_i)$ at time t_i. (3.12d) - (3.12f) are the state, output and the zone-tracking constraints, where $\mathbb{U}$, $\mathbb{Y}$, and $\mathbb{Y}_z$ are compact sets. The time-variant target zone $\mathbb{Y}_z(t_j)$ is a subset of $\mathbb{Y}$ and is defined as follow:

$$\mathbb{Y}_z(t_j) = \{y(t)|\underline{y}(t_j) \leq y(t_j) \leq \overline{y}(t_j)\}, \quad j = i, i+1, \cdots, i+N-1 \tag{3.13a}$$

$$\underline{y}(t_j) = \min\left[\underline{y}(t_i) \cdot e^{\mu \cdot \frac{t_j - t_i}{N}}, \underline{Y}\right] \tag{3.13b}$$

$$\overline{y}(t_j) = \max\left[\overline{y}(t_i) \cdot e^{-\mu \cdot \frac{t_j - t_i}{N}}, \overline{Y}\right] \tag{3.13c}$$

where $\underline{y}(t_j)$ and $\overline{y}(t_j)$ represent the lower and upper bound of the target zone at time step t_j. The boundaries of the target zone varies over time exponentially from their initial value

$\underline{y}(t_i)$ and $\overline{y}(t_i)$ until they are converged to the final value $\underline{Y}$ and $\overline{Y}$, where $\underline{Y} \leq \overline{Y}$. μ is a non-negative scalar that determines the shrinking speed of the boundaries. Larger μ leads to more aggressive shrinkage and vise versa. When $\mu = 0$, the target zone becomes time-invariant (i.e. $\underline{y}(t_j) = \underline{y}(t_i)$, $\overline{y}(t_j) = \overline{y}(t_i)$). In this chapter, $\underline{y}(t_i) = 0.18$ and $\overline{y}(t_i) = 0.23$ are used.

Note that the magnitude of the system input u is on the scale of 10^{-8} to 10^{-5}, which poses numerical issues for solving optimizations problems. Thus u is scaled before feeding to the optimization solver. The scaled input u_{scaled} takes values between 0 and 1. Unless specified otherwise, $Q = 4000$, $R = 100$ are employed.

$N = 20$ is used in this chapter. The sample time Δ of the optimization is 2 hours (i.e. $\Delta = 7200$s), meaning that at a given time instant i, the agro-hydrological dynamics in the next 40 hours will be optimized ($u^* = [u^*(t_i|t_i), u^*(t_i + 1|t_i), \cdots, u^*(t_i + N|t_i)]$). Following the receding horizon control strategy, only the optimal input at the first step $u^*(t_i|t_i)$ will be applied to the system. The optimization problem is solved repeatedly over the horizon of interest $N_{sim} = 60$, which optimizes the system dynamics over a 5 days duration.

3.6 ZMPC Simulation Results

ZMPC simulation results under various operating conditions are presented in this section. The same initial condition is employed for all simulations for fair comparisons. To be more specific, $x_0 = -0.2$ $[m]$ is the initial state at the rooting depth and is employed for the discretized Richards equation, while the corresponding system output $y_0 = 0.266$ $[m^3/m^3]$ is used for the NN models. First, in Section 3.6.1, the control performance of the proposed framework is compared with the discretized Richards equation and the single LSTM discussed in previous sections. Simulations are performed under the controller defined by (3.12) with a time-invariant target zone ($\mu = 0$) in the presence of process and/or measurement noises. Then the performance of the two-layer-NN-based ZMPC with a shrinking target zone in the

presence of significant noise and weather disturbance is presented in Section 3.6.2. The effects of tuning parameters μ, $\underline{Y}$, and $\overline{Y}$ in ZMPC performance are investigated. The basic ZMPC with a time-invariant target zone is employed as the benchmark controller for performance comparison.

3.6.1 Optimization performance validation

In this section the performance of the proposed framework in ZMPC applications is validated by comparing the simulation results with those obtained based on the discretized Richards equation and the single LSTM. Table 3.6 summarizes the average computational time for solving the optimization problem, the economic performance of ZMPC simulations, and the percentage difference in economic performance based on different models. The economic performance is measured by the accumulated irrigation amount I_T in millimeters, which is calculated as follows:

$$I_T = \sum_{t=0}^{N_{sim}} u(t) \cdot \Delta \cdot 1000$$

where $u(t)$ is the irrigation rate at time t with the unit m/s. The percentage difference in the accumulated irrigation is calculated with respect to the result obtained based on the Richards equation. The optimal control strategy and the corresponding output trajectories obtained based on different models are presented in Figure 3.10. The results shown in Table 3.6 and Figure 3.10 are obtained in the presence of 2% process noise. Figure 3.11 presents the optimal system input and output trajectories obtained with 5% process and 5% measurement noise based on the single LSTM and the proposed framework. The colors and markers employed in Figure 3.11 are the same as those in Figure 3.10. Table 3.7 summarizes more ZMPC simulation results based the single LSTM and the proposed framework in the presence of process and measurement noise.

As presented in Table 3.6, both the single LSTM and the proposed framework help to reduce the computational cost significantly. The computational cost reduction achieved by

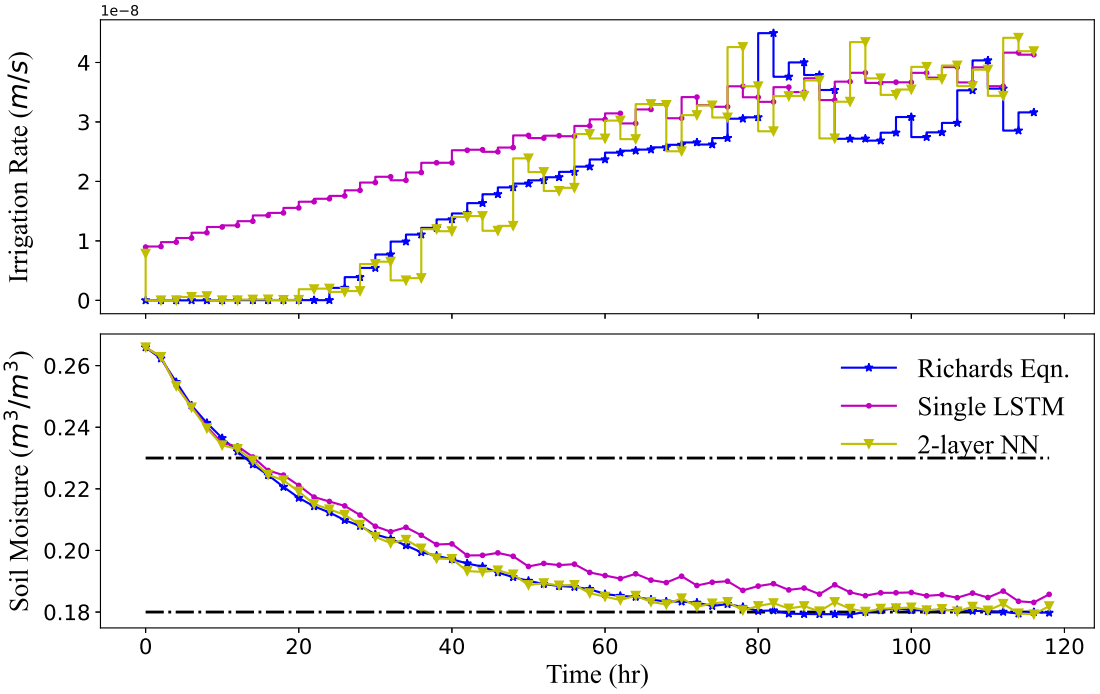

Figure 3.10: Optimal control strategy and the corresponding output response of ZMPC in the presence of 2% process noise. Discretized Richards equation (star marker), the single LSTM (dot marker), the two-layer NN framework (triangle marker).

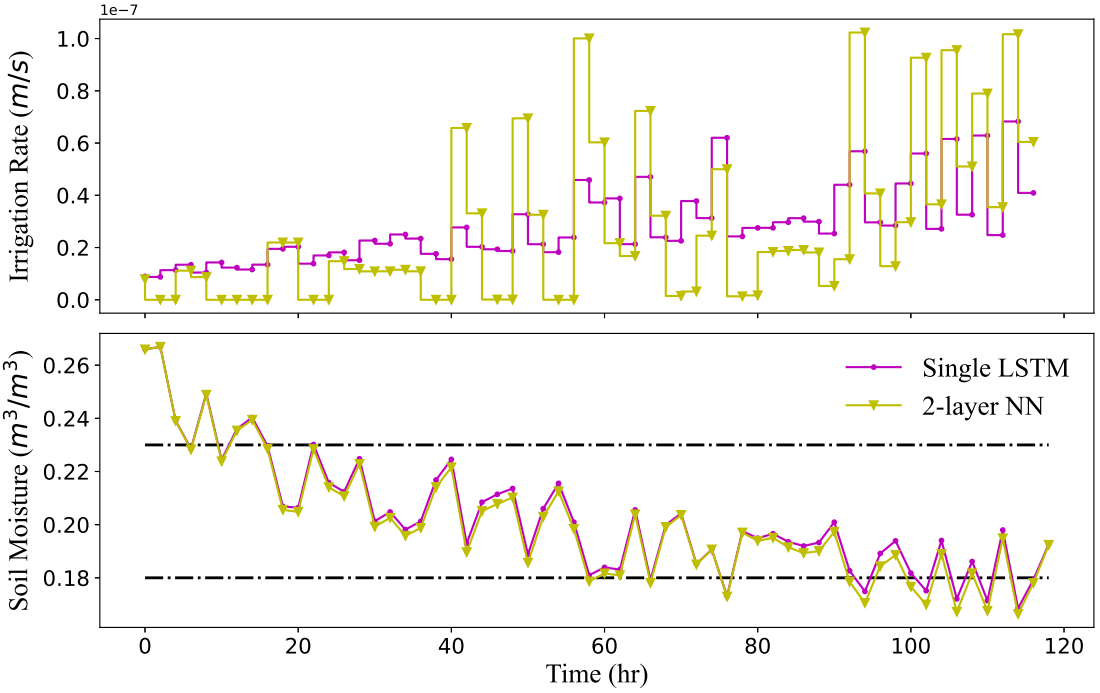

Figure 3.11: Optimal control strategy and the corresponding output response of ZMPC in the presence of 5% process noise and 5% measurement noise.

Table 3.6: ZMPC simulation results in the presence of 2% process noise.

Model	Avg. Time (s)	I_T (mm)	% Diff. in I_T
Richards Eqn.	493	8.33	-
Single LSTM	12.5	11.8	41.4
two-layer NN	35.4	9.05	8.66

Table 3.7: ZMPC simulation results in the presence of process noise (P.N.) and measurement noise (M.N.).

Noise	Model	I_T (mm)
2% M.N.	Single LSTM	11.6
	two-layer NN	8.74
5% P.N.	Single LSTM	12.0
	two-layer NN	9.54
5% M.N.	Single LSTM	12.0
	two-layer NN	9.91
2% P.N. & 2% M.N.	Single LSTM	11.7
	two-layer NN	9.46
5% P.N. & 5% M.N.	Single LSTM	12.1
	two-layer NN	11.2

the single LSTM is more significant compare to the two-layer NN due to the simplicity of its structure. With only 8.66% deviation, the economic performance of controller using the two-layer NN is similar to that using the Richards equation, which is significantly better compared to that obtained with the single LSTM. The trajectories in Figure 3.10 indicate the same, as the optimal control law and the system response of the two-layer NN and the Richards equations are highly overlapped. Based on the Richards equation, no irrigation

is applied for the first a few steps and starts to increase slowly over time. The two-layer NN leads to similar control strategy with the minor differences caused by the plant-model-mismatch. As for the single LSTM, the irrigation policy is less efficient, which is nonzero over the entire horizon. The output trajectories converge to the lower bound of the target zone for simulations based on the Richards equation and the proposed framework. The responses are reasonable as the economic objective is to reduce the amount of irrigation required. On the other hand, for the single-LSTM-based ZMPC, slower convergence and offsets with respect to the lower bound of the target zone is observed in the optimal output trajectory.

From Tables 3.6 and 3.7, it is observed that compared to process noise, measurement noise has stronger impacts on the simulation results. As shown in Table 3.7, except for the case that considered 5% process noise and 5% measurement noise, control with the two-layer NN has significantly better economic performances compared to that using the single LSTM. As shown in Figure 3.11, in the presence of 5% process noise and 5% measurement noise, the two-layer NN provides more aggressive control policy to capture the effect of the noise compare to the single LSTM. However, the impact of noise in this case is more significant compare to the control action due to the slow-responding nature of the agro-hydrological system, as the output trajectories obtained based on the two models are similar. To summarize, the proposed framework is shown to have better performances compared to the single LSTM in ZMPC applications. However, some minor violations of the target zone is observed at the end of the horizon of interest (Figure 3.11) due to the effect of noise, which is not ideal as low soil moisture may cause severe consequences in crop growing. Furthermore, precipitation is an essential disturbance in real-world applications and should be considered. In the following section, we investigate the performance of a shrinking target zone in capturing the effect of significant noise and weather disturbances.

Table 3.8: ZMPC simulation results with different $\underline{Y}$ and $\overline{Y}$

Target Zone	I_T (mm)	% Diff. - I_T	Zone Viol. (m^3/m^3)	% Diff. - Zone Viol.
Time Inv. Z.	6.16	-	0.007	-
$\underline{Y} = \overline{Y}, \mu = 1$	9.91	61.0	0.0005	93.1
$\underline{Y} < \overline{Y}, \mu = 1$	9.53	54.9	0.001	85.1

3.6.2 Shrinking zone v.s. time-invariant zone

In this section, the performance of the proposed shrinking-zone ZMPC and the ZMPC with time-invariant zone (referred to as the basic ZMPC in the following text) is compared in the presence of 5% measurement noise and weather disturbance. The controller is assumed to have information regarding the weather through weather forecasting with some uncertainties. In this chapter we consider 20% error in weather forecasting. Various parameter combinations are investigated. In Section 3.6.2.1, μ is kept constant and the effect of $\underline{Y}$ and $\overline{Y}$ discussed. The opposite is done in Section 3.6.2.2, where $\underline{Y}$ and $\overline{Y}$ are kept constant with varying μ.

3.6.2.1 Tuning $\underline{Y}$ and $\overline{Y}$

Figure 3.12 presents two sets of shrinking zones defined with different $\underline{Y}$ and $\overline{Y}$. The darker zone bounded by dashed boundaries has $\underline{Y} = \overline{Y} = 0.205$, meaning that the target zone shrinks to the center point of the original zone. The lighter zone bounded by dashed-dotted boundaries terminates to a zone with $\underline{Y} = 0.20$, $\overline{Y} = 0.21$. For both controllers, $\mu = 1$ is used. The ZMPC results are shown in Figure 3.13. The results obtained based on the basic ZMPC (i.e. $\underline{Y} = 0.18$, $\overline{Y} = 0.23$) is employed as the performance benchmark. The results obtained based on the two shrinking zone design are presented with the same line style as those used in Figure 3.12. The economic and zone-tracking performance of the ZMPC controllers are summarized in Table 3.8. The percentage difference of the total amount of irrigation and

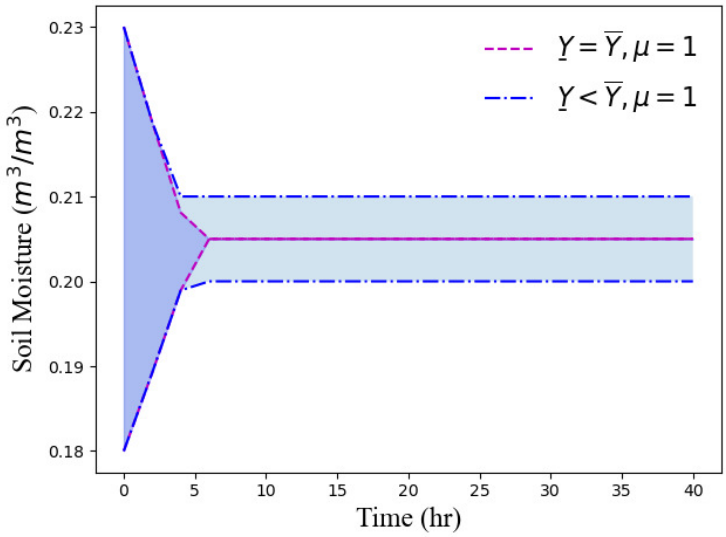

Figure 3.12: Shrinking zones with different $\underline{Y}$ and $\overline{Y}$.

zone-tracking performances are calculated based on the basic ZMPC controller.

From Figure 3.13, it is observed that with a time-invariant zone, the controller is able to capture the effect of the weather disturbance in general. When precipitation is foreseen, the irrigation amount is reduced. However, minor violation of the target zone at the lower bound is still observed due to the presence of noise. As presented in the figure, the usage of shrinking zone helps to avoid the violation. Instead of oscillating around the lower bound of the target zone, the output trajectories converge toward the center of the zone. This provides more robustness in the controller in the presence of significant noise. It is also noticed however, the employment of shrinking zones has significantly affect the total amount of irrigation. As summarized in Table 3.8, the two shrinking zones considered increase the accumulated irrigation amount by 61.0% and 54.9% respectively. Figure 3.13 and Table 3.9 indicate that the output responses based on the two shrinking zone designs are very similar, while the design that terminates to a zone performs better economically. Thus, in the following section we investigate the effect of μ with $\underline{Y} = 0.20$, $\overline{Y} = 0.21$.

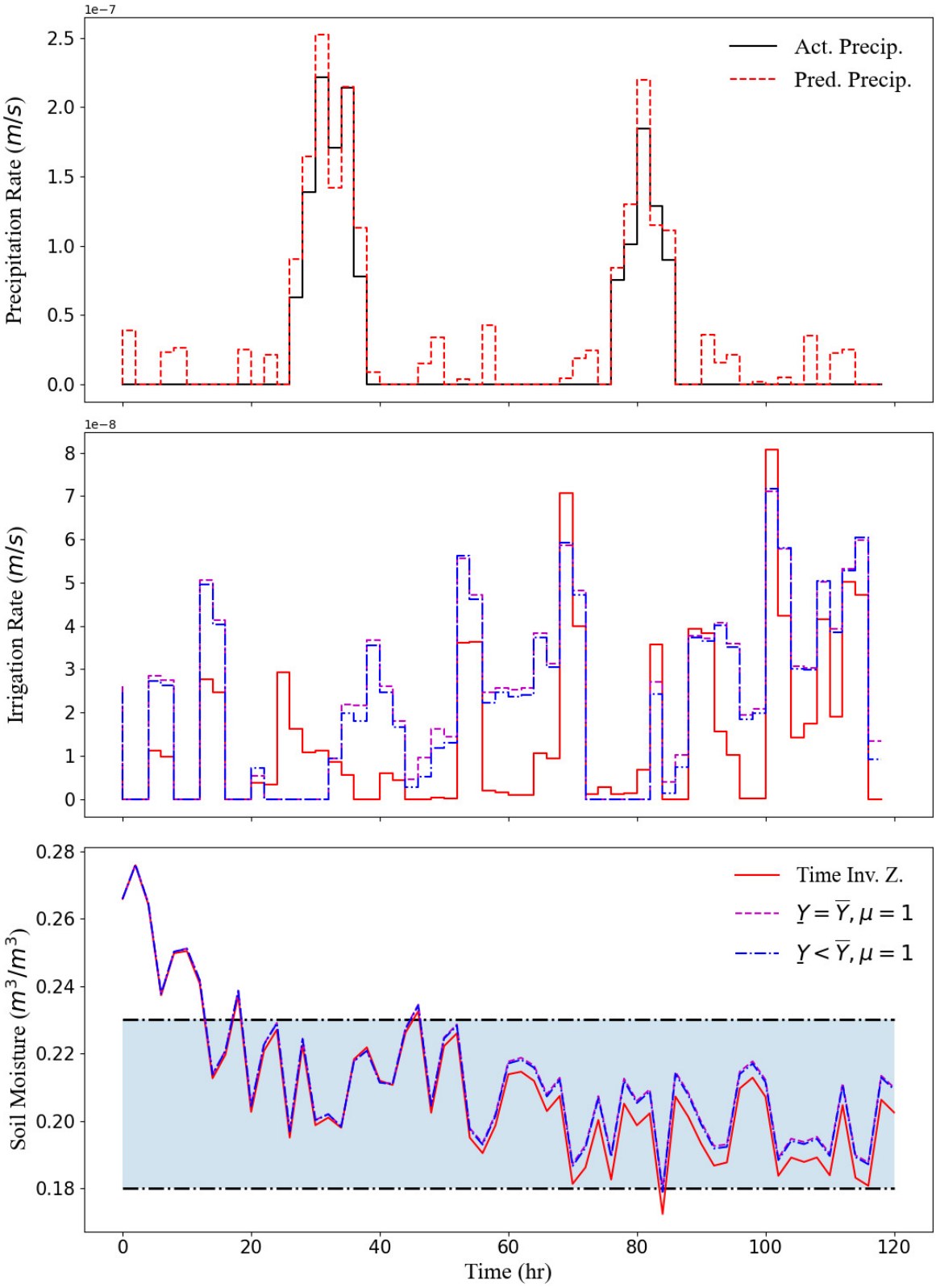

Figure 3.13: The precipitation data (top), optimal control strategies (middle), and corresponding output response (bottom) of ZMPC with different $\underline{Y}$ and $\overline{Y}$.

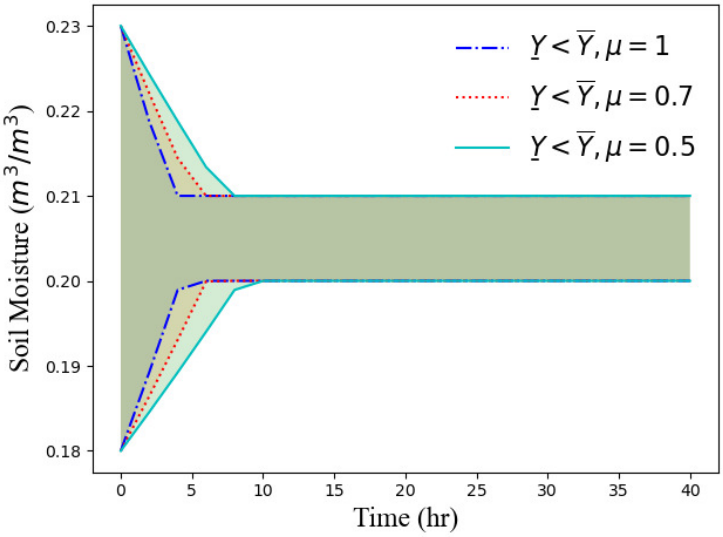

Figure 3.14: Shrinking zones with different μ.

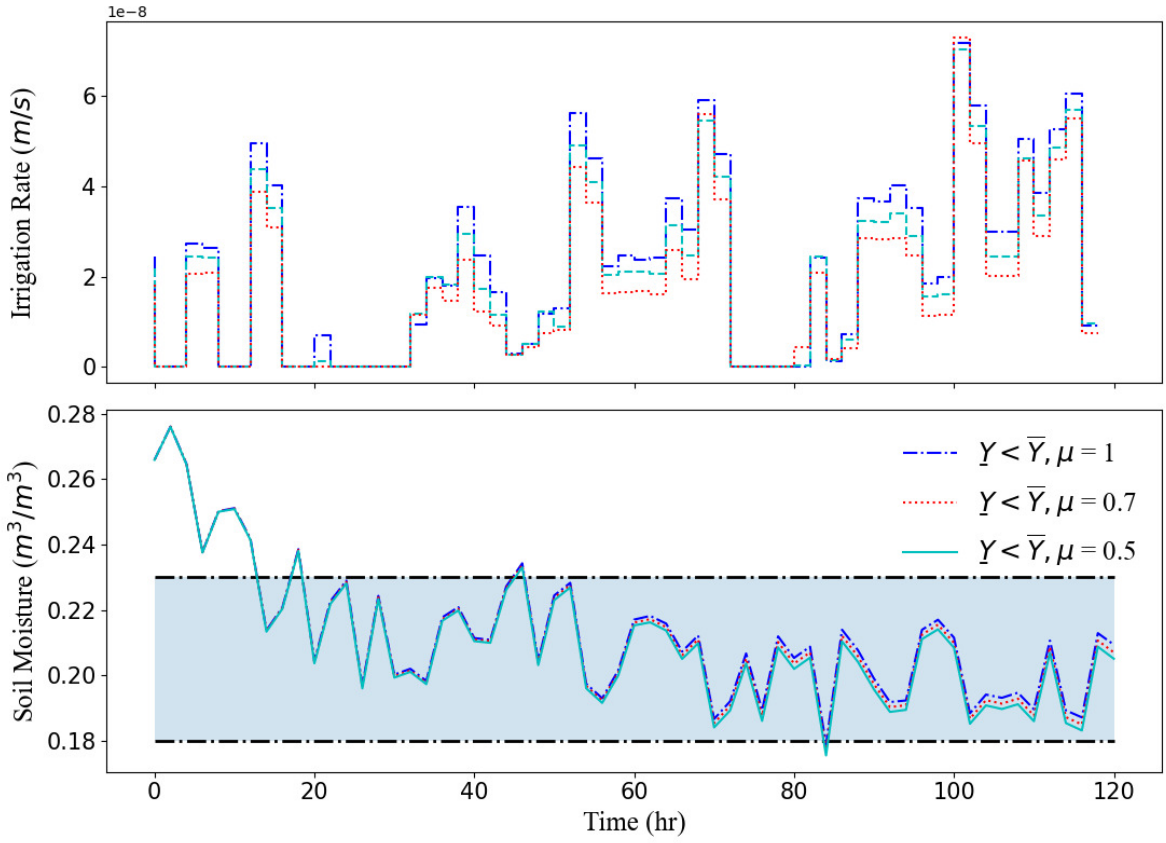

Figure 3.15: The optimal control strategies (top), and corresponding output response (bottom) of ZMPC with different μ.

Table 3.9: ZMPC simulation results with different μ

Target Zone	I_T (mm)	% Diff. - I_T	Zone Viol. (m^3/m^3)	% Diff. - Zone Viol.
Time Inv. Z.	6.16	-	0.007	-
$\underline{Y} < \overline{Y}, \mu = 1$	9.53	54.9	0.001	85.1
$\underline{Y} < \overline{Y}, \mu = 0.7$	8.45	37.2	0.003	62.0
$\underline{Y} < \overline{Y}, \mu = 0.5$	7.46	21.2	0.005	40.4

3.6.2.2 Tuning μ

Three shrinking zone designs with different μ are shown in Figure 3.14. $\mu = 1$ provides the fastest shrinking speed and thus the smallest zone. The second largest zone bounded by dotted lines has $\mu = 0.7$, while the largest zone with solid boundaries has $\mu = 0.5$. The shrinking process becomes less aggressive as μ decrease. Figure 3.15 shows the optimal control strategies and the corresponding output responses of the three ZMPC controllers. The same line styles as those used in Figure 3.14 are used in Figure 3.15. The total irrigation consumption of the three controllers is summarized in Table 3.9, where the result obtained with the basic ZMPC is again presented as the performance benchmark for consistency.

From Figure 3.15, it is observed that as the magnitude of μ decrease, the output response shifts towards the lower bound of the original target zone. The target zone violation thus becomes more significant, which is consistent with the results presented in Table 3.9. It is also noticed that with reducing μ value, the total irrigation consumption reduces. In summary, in the presence of noise and disturbances, a trade off exists between the amount of irrigation applied and the zone-tracking performance. Depending on the priority of the control objectives, parameters can be tuned accordingly. If some slight zone violation is allowed, a time-invariant target zone or a shrinking zone with smaller μ and greater difference between $\underline{Y}$ and $\overline{Y}$ can be employed for better economic performance. Vice versa, a shrinking zone that shrinks more significantly over the control horizon (i.e. larger μ and narrower

terminating zone) can be used if the zone-tracking objective is superior compared to the economic performance.

3.7 Summary

A two-layer NN framework is proposed in this chapter to approximate the agro-hydrological system of interest, which is shown to be superior to a single LSTM for both open-loop prediction and closed-loop control applications. Based on the two-layer NN model, a ZMPC strategy is proposed to maintain the soil moisture content at the root zone within a certain zone. To handle the effect of noise and weather disturbance, a target zone that shrinks along the control horizon is employed in the controller. The effects of different shrinking rates and widths of the terminal zone are discussed. Through simulations, it is noticed that steeper shrinking zones help to reduce the zone violation while significantly increasing the amount of irrigation required. Mild shrinkage in the target zone leads to lower economic costs but also minor zone violations. Depending on the priority of the control objective, the hyper-parameters can be tuned accordingly.

Chapter 4

Robust MPC with zone-tracking

In this chapter, we will investigate the theoretical stability aspect of MPC. In particular, we propose a robust ZMPC with guaranteed convergence in the presence of bounded disturbance, which is an extension of the results obtained in [41]. The proposed design rejects the effect of disturbance by modifying the target set. The modified target set is a subset of the actual target set, and the shrinkage is proved to be upper-bounded. A terminal constraint is employed to ensure the closed-loop stability, which forces the system state to converge to a forward control invariant set (CIS) inside the modified target set. Based on the stability theory, an algorithm for determining the modified target zone is proposed[3].

On top of the generalized robust ZMPC, we consider a secondary economic objective. Instead of introducing economic zones as proposed in [40], we prioritize the zone-tracking objective and establish stability based on the ZMPC alone. This implies that the economic objective is optimized only if the zone-tracking objective is satisfied, which provides a larger feasible region and more flexibility, as the economic objective can be modified without affecting the stability property of the controller.

The performance of the proposed ZMPC is validated through simulations based on a continuous stirred tank reactor (CSTR). The performance of the generalized ZMPC proposed in [41] is employed as the benchmark reference. To show the effectiveness of the proposed

[3]This chapter has been published as a journal paper.

design, simulations based on alternative controller formulations are performed and the results are compared with those obtained based on the proposed design. The effectiveness of the proposed algorithm for computing the modified target set is tested through simulations with different values of the tuning parameter. A remark regarding the selection of the tuning parameter is summarized.

4.1 Preliminaries

4.1.1 Notation

Throughout this chapter, $|\cdot|$ denotes the Euclidean norm of a scalar or a vector. The operator $||\cdot||_n$ denotes the n-norm of a scalar or a vector. $\mathbb{I}_M^N$ represents the set of integers from M to N $\{M, M+1, \cdots, N\}$. $\mathbb{I}_{\geq 0} = \{0, 1, 2, ...\}$ denotes the set of non-negative integers. For two sets A and B, the set subtraction is defined as $A \backslash B = \{a \in A | a \notin B\}$. The operators $\max\{\cdot\}$ and $\min\{\cdot\}$ find the maximum and minimum of each element in a vector variable respectively.

The operator $a \odot b$ represents element-wise multiplication between vectors a and b.

4.1.2 Description and problem formulation

The following discrete-time nonlinear system is considered in this chapter:

$$x(n+1) = f(x(n), u(n), w(n)) \tag{4.1}$$

where $x(\cdot) \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ denotes the state vector, $u(\cdot) \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ denotes the control input vector, and $w(\cdot) \in \mathbb{W} \subseteq \mathbb{R}^{n_w}$ denotes the disturbance vector. $n \in \mathbb{I}_{\geq 0}$ denotes the time step.

A few assumptions are enforced on the system of interest throughout this chapter and are listed below:

Assumption 1. *The constraint sets $\mathbb{X}$ and $\mathbb{U}$ are compact and coupled such that:*

$$(x(n), u(n)) \in \mathbb{Z} \subseteq \mathbb{X} \times \mathbb{U}$$

Furthermore, the sets can be expressed as following element-wise constraints:

$$\mathbb{X} := \left\{ x \middle| x_{lb} \leq x \leq x_{ub} \right\}, \ \mathbb{U} := \left\{ u \middle| u_{lb} \leq u \leq u_{ub} \right\}$$

where x_{lb}, x_{ub}, u_{lb}, and u_{ub} are constant vectors, with each elements being the element-wise lower and upper bound of the state and input. The expressions imply that the constraint sets are polyhedrons in the corresponding space bounded by hyperplanes.

Assumption 2. *The disturbance vector w is bounded and contains the origin in its domain:*

$$\mathbb{W} := \{ w \in \mathbb{R}^{n_w} : ||w||_\infty \leq \theta, \theta > 0 \} \tag{4.2}$$

such that the infinity norm of w is bounded by a positive constant θ.

Assumption 3. *The function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ is locally Lipschitz with respect to x and w for all $x \in \mathbb{X}$, $u \in \mathbb{U}, w \in \mathbb{W}$. This implies there exist positive constants L_x and L_w such that:*

$$|f(x, u, w) - f(z, u, 0)| \leq L_w |w| + L_x |x - z| \tag{4.3}$$

which indicates that the function f cannot change drastically as x and w changes.

The control objective of this chapter is to drive the state of system (4.1) into a tracking target set $\mathbb{X}_t$ and keeps it inside $\mathbb{X}_t$ thereafter in the presence of process uncertainties. The target set $\mathbb{X}_t$ can be expressed as the following element-wise constraints:

$$\mathbb{X}_t := \left\{ x \middle| x_{lb}^t \leq x \leq x_{ub}^t \right\} \tag{4.4}$$

where x_{lb}^t and x_{ub}^t are constant vectors, with each element being the element-wise lower and

upper bound on the state.

To achieve the control objective, we propose a robust ZMPC with guaranteed closed-loop stability in the Lyapunov sense, which is introduced in Section 4.2. The proposed robust ZMPC is an extension of the generalized ZMPC proposed in [41], which is introduced first in Section 4.1.3 for the completeness.

4.1.3 A generalized ZMPC formulation

The generalized ZMPC proposed in [41] is presented in this section. This generalized ZMPC assumes perfect knowledge regarding the system dynamics (i.e. $x(n+1) = f(x(n), u(n), 0)$). Throughout this chapter, we refer to this generalized ZMPC as the nominal ZMPC, which is used as the referencing benchmark for the performance validation of our proposed ZMPC. The generalized ZMPC formulation for time instant n is defined as follows:

$$V_N^0(x(n)) = \min_{u(0),\cdots,u(N-1)} \sum_{i=0}^{N-1} \ell_z(\hat{x}(i)) \tag{4.5a}$$

$$\text{s.t. } \hat{x}(i+1) = f(\hat{x}(i), u(i), 0), \quad i \in \mathbb{I}_0^{N-1} \tag{4.5b}$$

$$\hat{x}(0) = x(n) \tag{4.5c}$$

$$\hat{x}(i) \in \mathbb{X}, \quad i \in \mathbb{I}_0^{N-1} \tag{4.5d}$$

$$u(i) \in \mathbb{U}, \quad i \in \mathbb{I}_0^{N-1} \tag{4.5e}$$

$$\hat{x}(N) \in \mathbb{X}_f \tag{4.5f}$$

where (4.5b) is the nominal process model constraint, (4.5c) is the initial condition constraint, and (4.5d) and (4.5e) are the state and input constraints, respectively. (4.5f) denotes the terminal constraint, where $\mathbb{X}_f$ denotes the terminal set. N is a positive constant that represents the control horizon of the controller. $\hat{x}(i)$ denotes the predicted states over the prediction

horizon for $i \in \mathbb{I}_0^{N-1}$. $\ell_z(\cdot)$ represents the zone-tracking objective and is defined as follows:

$$\ell_z(z) = \min_{z_z} c_1||z - z_z||_1 + c_2||z - z_z||_2^2 \tag{4.6a}$$

$$s.t.\ z_z \in \mathbb{X}_t \tag{4.6b}$$

where c_1 and c_2 are non-negative weighting factors, $\mathbb{X}_t \subset \mathbb{X}$ denotes the target zone, and z_z is a slack variable that is forced to stay inside the target zone by (4.6b). The zone-tracking objective is a weighted summation of the 1-norm and squared 2-norm of the minimum difference between the actual state and the slack variable, which is a representation of the distance between the system state x and the target set $\mathbb{X}_t$. When the system state is outside the target set, ℓ_z is positive. When the system state converges to the target set, ℓ_z equals to zero.

Taking the definition of (4.6) into consideration, the objective of the controller (4.5) is to find the optimal input sequence for N future steps such that the system state is driven towards and kept inside the target zone while constraints (4.5b) - (4.5e) are satisfied. In addition, the terminal state x_N is forced to converge to $\mathbb{X}_f \subseteq \mathbb{X}_t^M \subseteq \mathbb{X}_t$, where $\mathbb{X}_t^M$ denotes the largest forward control invariant set (CIS) inside $\mathbb{X}_t$. The definition of a forward CIS is provided as follows:

Definition 1. *(Forward control invariant set [68]) A set $\mathbb{X}_r \subseteq \mathbb{X}$ is said to be a forward control invariant set (CIS) for the system $x(n+1) = f(x(n), u(n), 0)$ if there exists a control policy $u(n) = \mu(x(n))$ for every $x(n) \in \mathbb{X}_r$ such that $x(n+1) = f(x(n), \mu(x(n)), 0) \in \mathbb{X}_r$.*

We refer to the forward CIS as CIS hereafter for simplicity. The nominal ZMPC is proved to be closed-loop stable in the Laypunov sense when the system model is exactly known. In this chapter, we extend the design of the nominal ZMPC to consider nonlinear systems with model-plant mismatch in the form of process disturbance as shown in (4.1). A robust ZMPC with guaranteed closed-loop stability is proposed in the following section.

Remark 1. *Recall from the previous session, the objective of an ZMPC is to keep the state*

inside the target set $\mathbb{X}_t$, while in (4.5f), the state is forced to converge into a forward CIS $\mathbb{X}_f$. It is worth mentioning that if Assumptions 1 and 3 hold, the state converges into $\mathbb{X}_t$ asymptotically if and only if the state converges into the terminal set $\mathbb{X}_f$, which is proved in [41]. Thus, the terminal constraint (4.5f) ensures the closed-loop stability of the generalized ZMPC (4.5).

4.2 The Proposed Robust ZMPC

The proposed robust ZMPC is presented in this section. The effect of process uncertainty is encountered through modifying the tracking target set and the terminal set. The proposed robust ZMPC is designed based on the nominal system $\tilde{x}(n+1) = f(x(n), u(n), 0)$, which predicts the nominal state one-step into the future based on the known actual state at any given time $n \in \mathbb{I}_{\geq 0}$. The value of the disturbance $w(n)$ is unknown and is assumed to be 0 in the nominal model. The proposed ZMPC design for time instant n is presented as follows:

$$V_N^0(x(n)) = \min_{u(0),\cdots,u(N-1)} \sum_{i=0}^{N-1} \tilde{\ell}_z(\tilde{x}(i)) \tag{4.7a}$$

$$\text{s.t. } \tilde{x}(i+1) = f(\tilde{x}(i), u(i), 0), \quad i \in \mathbb{I}_0^{N-1} \tag{4.7b}$$

$$\tilde{x}(0) = x(n) \tag{4.7c}$$

$$\tilde{x}(i) \in \mathbb{X}, \quad i \in \mathbb{I}_0^{N-1} \tag{4.7d}$$

$$u(i) \in \mathbb{U}, \quad i \in \mathbb{I}_0^{N-1} \tag{4.7e}$$

$$\tilde{x}(N) \in \tilde{\mathbb{X}}_f \tag{4.7f}$$

where $\tilde{\ell}_z(\cdot)$ is defined as follows:

$$\tilde{\ell}_z(z) = \min_{z_z} c_1||z - z_z||_1 + c_2||z - z_z||_2^2 \tag{4.8a}$$

$$s.t.\ z_z \in \tilde{\mathbb{X}}_t \tag{4.8b}$$

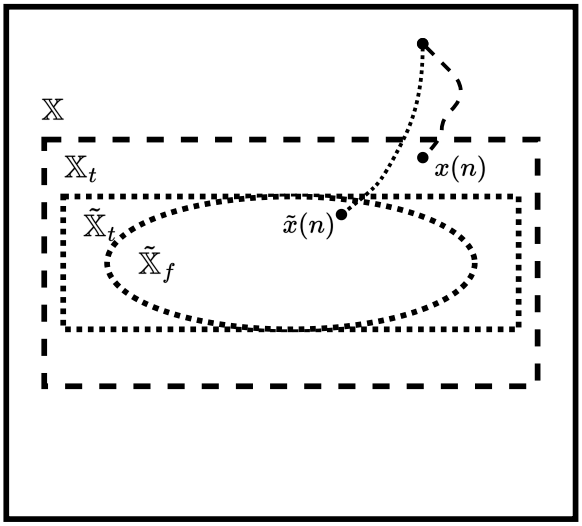

Figure 4.1: Potential nominal state trajectory (dotted line) and actual state trajectory (dashed line) of the closed-loop system under the proposed ZMPC.

Similar notations employed in (4.5) and (4.6) are adopted in the proposed controller (4.7) and (4.8) except that $\tilde{x}$ is used to represent the nominal state predicted by the nominal model. Compare to the nominal ZMPC, the key changes took place in the proposed design are the modified target set $\tilde{\mathbb{X}}_t$ and the terminal set $\tilde{\mathbb{X}}_f$ in (4.8b) and (4.7f), respectively. The modified target set is a subset of the original target set ($\tilde{\mathbb{X}}_t \subset \mathbb{X}_t$). The terminal constraint is needed to ensure the closed-loop stability with a CIS inside the modified target zone $\tilde{\mathbb{X}}_t$.

The proposed ZMPC is applied to system (4.1) in the receding horizon fashion. At each time instant $n \in \mathbb{I}_{\leq 0}$, the optimal input $u(i|n)^*,\ i \in \mathbb{I}_0^{N-1}$ for N future steps is obtained by solving the optimization problem (4.7). Only the optimal input for $i = 0$ is applied to the system at time n. At time $n + 1$, the optimization problem (4.7) is evaluated again with updated initial condition.

Figure 4.1 presents potential trajectories of the closed-loop nominal state and the actual state based on the proposed robust ZMPC. The solid box represents the state space $\mathbb{X}$, while the rectangles bounded by the dashed and the dotted lines represent the actual target set $\mathbb{X}_t$ and the modified target set $\tilde{\mathbb{X}}_t$, respectively. The ellipsoid bounded by the dotted line denotes the largest CIS $\tilde{\mathbb{X}}_t^M$ inside the modified target set, which is used as the terminal set

$\tilde{\mathbb{X}}_f$ of the proposed controller. Oftentimes, a nominal ZMPC tends to drive the system state to the boundary of the target set, which likely leads to target set violation in the presence of process uncertainties. By ensuring the closed-loop convergence of the nominal state to the modified target set, which is a subset of the actual target set, a buffer zone is provided to reject the effect of process uncertainties. The terminal constraint (4.7f) ensures that the nominal state will stay inside the control invariant terminal set $\tilde{\mathbb{X}}_f$ and thus $\tilde{\mathbb{X}}_t$ once it has first converged. More details regarding the closed-loop stability of the proposed controller are discussed in the following section.

Remark 2. *Throughout the manuscript, we use $x(\cdot)$ to denote the true system state at a given time step. Contrarily, $\tilde{x}(\cdot)$ denotes the system state predicted based on the knowledge available to the controller. Following similar ideas, the set $\mathbb{X}_t$ reflects the true zone control objectives we have on the system of interest, which is referred to as the actual target set. For example, the objective is to keep the reactor temperature in a certain range. The set $\tilde{\mathbb{X}}_t$ in comparison, is mathematically determined based on the actual target set with no physical significance and is referred to as the modified target set. Furthermore, we define the set $\mathbb{X}_t^M$ and $\tilde{\mathbb{X}}_t^M$ as the largest CIS inside $\mathbb{X}_t$ and $\tilde{\mathbb{X}}_t$, respectively.*

4.3 Stability Analysis

In this section, we investigate the closed-loop stability of the proposed robust ZMPC (4.7). First, we introduce the definition of the N-step reachable set, which is essential in characterizing the feasibility of the proposed robust ZMPC.

Definition 2. *(N-step reachable set [41]) We use $\mathbb{X}_N(\mathbb{X}, \mathbb{X}_f)$ to denote the set of states in $\mathbb{X}$ that can be steered to $\mathbb{X}_f$ in N steps while satisfying the state and input constraints $(x, u) \in \mathbb{Z}$. That is,*

$$\mathbb{X}_N(\mathbb{X}, \mathbb{X}_f) = \{x(0) \in \mathbb{X} \mid \exists\, (x(n), u(n)) \in \mathbb{Z}, n \in \mathbb{I}_0^{N-1}, x(N) \in \mathbb{X}_f\}$$

The following assumption is employed to ensure the feasibility of the proposed robust ZMPC.

Assumption 4. *The terminal set $\tilde{\mathbb{X}}_f$ and the N-step reachable set $\mathbb{X}_N(\mathbb{X}, \tilde{\mathbb{X}}_f)$ are compact and nonempty.*

For the completion of this chapter, here we summarize the essential results obtained in [41] regarding the nominal ZMPC as Theorem 1.

Theorem 1. *(Stability of the nominal ZMPC [41]) If Assumptions 1 and 4 hold, $x_0 \in \mathbb{X}_N(\mathbb{X}, \mathbb{X}_f)$, $w(n) = 0$, and in addition, the optimal value function $V_N^0(x(n))$ is locally continuous on $\mathbb{X}_t$, then: (i) the zone MPC is recursively feasible with $x(n) \to \mathbb{X}_t^M$, (ii) the terminal set $\mathbb{X}_t^M$ is asymptotically stable, (iii) the following inequality holds:*

$$V_N^0(\tilde{x}(n+1)) - V_N^0(\tilde{x}(n)) \leq -\ell_z(\tilde{x}(n)), \quad n \in \mathbb{I}_{\geq 0} \tag{4.9}$$

where V_N^0 is the optimum control objective function and is a Lyapunov function with respect to the set $\mathbb{X}_t^M$, and $\ell_z(\cdot)$ denotes the zone-tracking objective function defined in (4.6).

As the proposed ZMPC is applied to the system of interest in the receding horizon fashion, the optimization problem is solved at each time step with the updated information regarding the actual state at the initial time step over the prediction horizon (i.e. $\tilde{x}(n) = x(n)$). This implies only the one-step-ahead deviation caused by the plant-model mismatch between the nominal model and the actual model needs to be considered. The following Proposition provides an upper bound on the one-step-ahead deviation of the predicted state from the actual state:

Proposition 1. *(c.f. [40]) Consider the actual system (4.1) and the nominal system with $w(n) = 0$. If Assumption 2 is satisfied, starting from a known initial condition $x(n)$, the deviation of the actual state $x(n+1)$ from the predicted state $\tilde{x}(n+1)$ in one time step is bounded:*

$$|x(n+1) - \tilde{x}(n+1)| \leq \sqrt{n_x} L_w \theta, \ \forall x(n+1), \ \tilde{x}(n+1) \in \mathbb{X} \tag{4.10}$$

Based on the upper bound of the one-step-ahead deviation in the state, Proposition 2 provides an upper bound on the progression of the Lyapunov function $V_N^0(\cdot)$ in one time step from $x(n)$ to $x(n+1)$.

Proposition 2. *Consider the Lyapunov function $V_N^0(x)$. There exist positive constants K_V and H such that:*

$$V_N^0(x(n+1)) - V_N^0(x(n)) \leq -\tilde{\ell}_z(\tilde{x}(n)) + f_V(\sqrt{n_x}L_w\theta) \tag{4.11}$$

where n_x and θ are known parameters, $\tilde{\ell}_z(\cdot)$ denotes the zone-tracking objective function defined in (4.8), and f_V is a function defined as follows:

$$f_V(x) = K_V x + Hx^2$$

where K_V and H are positive constants.

Proof. Upon applying Taylor expansion to $V_N^0(x)$ around the predicted state $\tilde{x}(n+1)$, the following relation can be obtained:

$$\begin{aligned} &V_N^0(x(n+1)) = V_N^0(\tilde{x}(n+1)) + \\ &\left.\frac{\partial V_N^0}{\partial x}\right|_{\tilde{x}(n+1)} |x(n+1) - \tilde{x}(n+1)| + H.O.T. \end{aligned} \tag{4.12}$$

where $H.O.T.$ includes the higher order terms of the Taylor expansion. A positive constant H can be found for $x \in \mathbb{X}$ such that the following holds:

$$H.O.T. \leq H|x(n+1) - \tilde{x}(n+1)|^2 \tag{4.13}$$

Consider (4.12) and (4.13) correspondingly, we can obtain the following inequality regarding

the difference in V_N^0 due to the effect of disturbance in one step:

$$\begin{aligned} V_N^0(x(n+1)) &\leq V_N^0(\tilde{x}(n+1)) \\ &+ \left.\frac{\partial V_N^0}{\partial x}\right|_{\tilde{x}(n+1)} |x(n+1) - \tilde{x}(n+1)| \\ &+ H|x(n+1) - \tilde{x}(n+1)|^2 \end{aligned} \tag{4.14}$$

If Assumptions 3 and 4 are satisfied, there exists a positive constant K_V such that the magnitude of the partial derivative $\frac{\partial V_N^0}{\partial x}$ is bounded by it:

$$\left|\frac{\partial V_N^0}{\partial x}\right| \leq K_V \tag{4.15}$$

Define $f_V(x) = K_V x + Hx^2$. Based on (4.14) and (4.15), applying Proposition 1, the following relationship can be derived:

$$V_N^0(x(n+1)) \leq V_N^0(\tilde{x}(n+1)) + f_V(\sqrt{n_x} L_w \theta) \tag{4.16}$$

Taking into account of (4.9) and (4.16), plus the fact that $x(n) = \tilde{x}(n)$, the relationship in (4.11) can be obtained and this proves Proposition 2. ■

To derive the amount of shrinkage required in the target set that ensures the closed-loop convergence of the system in the presence of disturbance, we introduce the following definition of Lyapunov function level set:

Definition 3. *(Level set of Lyapunov function [40]) The set Ω_ρ is defined to be the level set of the Lyapunov function $V_N^0(\cdot)$:*

$$\Omega_\rho := \{x \in \mathbb{X} : V_N^0(x) \leq \rho\} \tag{4.17}$$

A Lyapunov function can be considered as a generalized energy function of the system of interest. Thus, the level set of a Lyapunov function is a set that contains all the states such

that the system energy is less than or equal to ρ.

Let $\Omega_{\rho_{max}}$ be the largest level set of V_N^0 inside $\mathbb{X}_t^M$:

$$\rho_{max} := max\{\rho : x \in \mathbb{X}_t^M, V_N^0(x) \leq \rho\} \tag{4.18}$$

Define $\Omega_{\rho_e} \subset \Omega_{\rho_{max}}$ that satisfies:

$$\rho_e = \rho_{max} - f_V(\sqrt{n_x} L_w \theta) \tag{4.19}$$

Assumption 5. *Ω_{ρ_e} is nonempty.*

Theorem 2. *Consider system (4.1) under the control of the proposed ZMPC (4.7). If $x(n) \in \Omega_{\rho_e}$ and Assumptions 3 - 5 are satisfied, then it is guaranteed that $x(n+1) \in \Omega_{\rho_{max}}$.*

Proof. At a given time instant n, with the initial condition $x(n) \in \Omega_{\rho_e}$, it is guaranteed that there exists a feasible control policy such that the nominal state at the next time step $\tilde{x}(n+1) \in \Omega_{\rho_e}$, as Ω_{ρ_e} is control invariant. Take the inequality expression (4.16) into consideration, the deviation in the value of Lyapunov function V_N^0 from $\tilde{x}(n+1)$ to $x(n+1)$ is bounded by $f_V(\sqrt{n_x} L_w \theta)$. Recall that Ω_{ρ_e} is a shrunk Lyapunov function level set with respect to $\Omega_{\rho_{max}}$, with the shrinkage defined in the Lyapunov function value by $f_V(\sqrt{n_x} L_w \theta)$, which proves Theorem 2. ■

Theorem 2 indicates the convergence of the actual state into a control invariant subset of the actual target set $\mathbb{X}_t$ if the initial state $x(n) \in \Omega_{\rho_e}$. The following theorem provides statements on the upper bound of the shrinkage amount in the target set required in the proposed ZMPC design such that the state is guaranteed to be driven into Ω_{ρ_e}.

Definition 4. *(The smallest polyhedron set around a known set) We define the smallest polyhedron set bounded by orthogonal hyperplanes $\mathbb{P}(\mathbb{S})$ around a given set $\mathbb{S} \subseteq \mathbb{R}^{n_s}$ as follows:*

$$\mathbb{P}(\mathbb{S}) = \{s | s_{lb} \leq s \leq s_{ub}\} \tag{4.20}$$

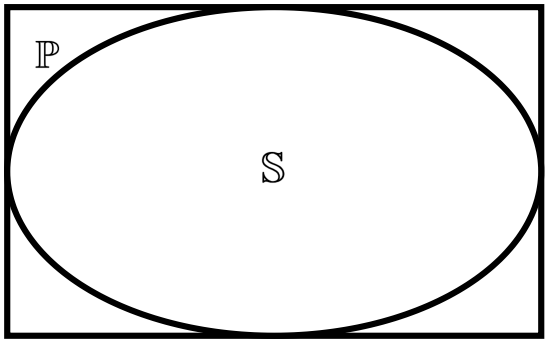

Figure 4.2: A sample diagram of $\mathbb{P}(\mathbb{S})$.

where $s_{lb}, s_{ub} \in \mathbb{R}^{n_s}$ *are vectors with each element* $s^i_{lb},\ i = 1, 2, \cdots, n_s$ *and* $s^i_{ub},\ i = 1, 2, \cdots, n_s$ *being the lower and upper bound of each dimension of the set* $\mathbb{S}$*:*

$$s^i_{lb} = \min\{s^i | s \in \mathbb{S}\},\ i = 1, 2, \cdots, n_s \tag{4.21a}$$

$$s^i_{ub} = \max\{s^i | s \in \mathbb{S}\},\ i = 1, 2, \cdots, n_s \tag{4.21b}$$

where s^i *is the* i*-th element of the vector* s*. A possible representation of the smallest polyhedron set* $\mathbb{P}(\mathbb{S})$ *and the corresponding set* $\mathbb{S}$ *is presented in Figure 4.2.*

Theorem 3. *Consider system (4.1) under the control of the proposed ZMPC (4.7). If Assumptions 1 - 5 are satisfied,* $x(0) \in \mathbb{X}_N(\mathbb{X}, \tilde{\mathbb{X}}_f)$*, and the following relationship holds:*

$$-\tilde{\ell}_z(x) + f_V(\sqrt{n_x} L_w \theta) < 0,\ \forall x \in \mathbb{X}_N(\mathbb{X}, \tilde{\mathbb{X}}_f) \backslash \Omega_{\rho_e} \tag{4.22}$$

then there exists $\overline{\varepsilon}, \underline{\varepsilon} \in \mathbb{R}^{n_x}$ *with non-negative elements that define the modified target zone* $\tilde{\mathbb{X}}_t$ *as follows:*

$$\tilde{\mathbb{X}}_t = \{x : \tilde{x}^t_{lb} \leq x \leq \tilde{x}^t_{ub} | \tilde{x}^t_{lb} - x^t_{lb} \leq \underline{\varepsilon}, x^t_{ub} - \tilde{x}^t_{ub} \leq \overline{\varepsilon}\} \tag{4.23}$$

such that the proposed ZMPC controller is recursively feasible and guarantees the closed-loop convergence of the system state into the actual target zone $\mathbb{X}_t$*. Note that* x^t_{lb} *and* x^t_{ub} *are known vectors containing the element-wise lower and upper bound of the actual target set* $\mathbb{X}_t$ *as defined in Assumption 1.*

Proof. We define the modified target set as the smallest polyhedron set around Ω_{ρ_e} (i.e. $\tilde{\mathbb{X}}_t = \mathbb{P}(\Omega_{\rho_e})$) and the corresponding terminal set $\tilde{\mathbb{X}}_f$ as the largest CIS inside $\tilde{\mathbb{X}}_t$. If Assumption 5 holds, it means there exists a valid set of $\tilde{x}^t_{lb}$ and $\tilde{x}^t_{ub}$ such that $\tilde{\mathbb{X}}_t \subset \mathbb{X}_t$ is bounded and nonempty. Recall that the actual target set $\mathbb{X}_t$ is bounded, meaning that the elements in vectors $\overline{\varepsilon}, \underline{\varepsilon} \in \mathbb{R}^{n_x}$ are bounded and non-negative.

Take the relationship defined by (4.22) into consideration. The value of the Lyapunov function V_N^0 is guaranteed to be decreasing until the state converges to Ω_{ρ_e}, thus ensuring the convergence of the nominal state $\tilde{x}$ into Ω_{ρ_e}.

In the presence of disturbance, the actual state x may deviate from Ω_{ρ_e} after the first convergence of the nominal state $\tilde{x}$. Taking Theorem 2 into consideration, it is guaranteed that with $x(n) \in \Omega_{\rho_e}$, $x(n+1) \in \Omega_{\rho_{max}}$, meaning the actual state is kept inside $\Omega_{\rho_{max}} \subset \mathbb{X}_t^M \subset \mathbb{X}_t$ in one step. As $\Omega_{\rho_e} \subset \Omega_{\rho_{max}} \subset \mathbb{X}_N(\mathbb{X}, \tilde{\mathbb{X}}_f)$, the actual state that has deviated in one step due to the effect of disturbance is guaranteed to be driven back to Ω_{ρ_e} in finite steps as (4.22) holds. This indicates that the proposed ZMPC controller is able to drive the system state into the actual target set $\mathbb{X}_t$ and keeps it inside. ■

Remark 3. *By definition, any CIS inside the modified target set $\tilde{\mathbb{X}}_t$ can be used as the terminal set $\tilde{\mathbb{X}}_f$. However, in order to gain a controller with a larger feasibility region and more degrees of freedom, we define $\tilde{\mathbb{X}}_f = \tilde{\mathbb{X}}_t^M$. The largest CIS inside the modified target set is approximated using the algorithm proposed in [69]. The algorithm provides an inner approximation of the largest CIS in the form of a polyhedron bounded by hyperplanes.*

4.4 An Algorithm for Estimating the Modified Target Set $\tilde{\mathbb{X}}_t$

The stability proof in the previous section ensures the existence of an upper bound of the shrinkage amount on the actual target set, however, it is very challenging to obtain explicit expressions of the Lyapunov function and numerical values of the parameters. In order to

make the proposed controller more practical in applications, we propose an algorithm for estimating and tuning the modified target set.

The proposed algorithm is inspired by a study on the problem in the context of linear systems. Consider a linear system as follows:

$$x(n+1) = Ax(n) + Bu(n) + Ew(n) \tag{4.24}$$

where A, B, and E are matrices of the appropriate dimensions. Let us assume that the linear system is controllable and there exists the following quadratic Lyapunov function for the system:

$$V_N^0(x) = x^T P x \tag{4.25}$$

where P is a symmetric positive definite matrix. Recall the Lipschitz continuity Assumption 3. If we substitute the linear system model (4.24) into the left hand side of the inequality (3), the following expression can be obtained:

$$|f(x,u,w) - f(x,u,0)| = |Ew| \leq L_w|w| \tag{4.26}$$

By the property of norm, the following expression holds:

$$|Ew| \leq |E||w| \tag{4.27}$$

Thus, we may define $L_w = |E|$ such that Assumption 3 holds. Recall the inequality relation (4.10):

$$|x(n+1) - \tilde{x}(n+1)| \leq \sqrt{n_x} L_w \theta$$

where n_x is the dimension of the state, θ is the upper bound of the one-norm of the disturbance and is assumed to be known. Thus for the linear system, the right-hand side of the inequality can be determined explicitly. By calculating the term $\sqrt{n_x} L_w \theta$, a conserva-

tive approximation of the norm of the one-step-ahead difference between the actual state $x(n+1)$ and the nominal state $\tilde{x}(n+1)$ in the presence of disturbance is obtained. For the nonlinear system of interest, the matrix E is essentially the sensitivity matrix of the state with respect to the disturbance (i.e. $\frac{\partial x}{\partial w}$). The value of the sensitivity matrix is a function of the state, input, and disturbance and can be calculated explicitly with a given combination of the variables.

Practically, based on inequality (4.10), we propose the following definition of the shrinkage amount s in the state boundaries of the target zone for the nonlinear system:

$$s = \gamma x^d_{max} \tag{4.28}$$

where $\gamma \in \mathbb{R}^+$ is a risk factor that helps to tune how conservative the proposed controller is, and x^d_{max} essentially estimates the largest potential effect the disturbance can have on the actual state compared to the nominal state in one-step-ahead prediction, of which the definition is inspired by (4.10). x^d_{max} can be calculated using Algorithm 3 based on sensitivity analysis.

In Algorithm 3, the set $\mathbb{X}^d$ contains $2^{n_z+n_w}$ number of vectors. Each vector represents the amount of accumulated effects of the disturbance w have on the state x at one of the extremes of the joint space $\mathbb{X}^M_t \times \mathbb{U} \times \mathbb{W}$. It is worth pointing out that the sensitivity matrix is computed for states inside the maximum CIS $\mathbb{X}^M_t$, that is, inside the actual target set. This is because it has been proved in [41] that for the nominal system, guaranteed convergence into the actual target set $\mathbb{X}_t$ is equivalent to the convergence into its maximum control invariant subset $\mathbb{X}^M_t$. Since the purpose of modifying the target set is to avoid zone violation once it has converged, we focus on rejecting the effect of disturbance when $x \in \mathbb{X}^M_t$. Furthermore, the effect of disturbance is only investigated for the individual state variables that have a zone-tracking objective on them, as it is common that the zone-tracking objective is not enforced on all state variables.

Algorithm 3 Estimation of the state deviation due to the presence of disturbance

Ensure: $x \in \mathbb{X}_t^M$, $u \in \mathbb{U}$, $w \in \mathbb{W}$

$I_t \leftarrow [\,]$

for $i = 0 : n_x - 1$ **do**

  if $x_{lb} < x_{lb}^t$ or $x_{ub} > x_{ub}^t$ **then**

    $I_t \leftarrow [I_t, 1]$

  else $I_t \leftarrow [I_t, 0]$

  end if

end for

Define $z \in \mathbb{R}^{n_z} \leftarrow [x, u]$, $n_z = n_x + n_u$

Determine the sensitivity function $\frac{\partial x}{\partial w} : \mathbb{R}^{n_z + n_w} \rightarrow \mathbb{R}^{n_x \times n_w}$

Find and store the maximum and minimum of each variables

$$z_{max} \in \mathbb{R}^{n_z} \leftarrow \max\{z\},\ z_{min} \in \mathbb{R}^{n_z} \leftarrow \min\{z\}$$

$$w_{max} \in \mathbb{R}^{n_w} \leftarrow \max\{w\},\ w_{min} \in \mathbb{R}^{n_w} \leftarrow \min\{w\}$$

Calculate the set of the accumulated sensitivity $\mathbb{X}^d$ for the state variables with zone-tracking objective at each extreme point of z and w:

$$\mathbb{X}^d := \left\{ \left[\frac{\partial x}{\partial w}\Big|_{z^i, w^j} \cdot w \right] \odot I_t \;\middle|\; z^i \in [z_{max}^i, z_{min}^i], i \in \mathbb{I}_0^{n_z - 1}, w^j \in [w_{max}^j, w_{min}^j], j \in \mathbb{I}_0^{n_w - 1}, w = [w^0, \cdots, w^{n_w - 1}] \right.$$

Find the element x_{max}^d with the largest norm in $\mathbb{X}^d$

$$x_{max}^d \leftarrow \left\{ x_{max}^d \;\middle|\; x_{max}^d \in \mathbb{X}^d, |x_{max}^d| \geq |x^d|, \forall x^d \in \mathbb{X}^d \right\}$$

return x_{max}^d

The modified target zone can be calculated based on the shrinkage amount s as follows:

$$\tilde{\mathbb{X}}_t := \left\{ x \middle| x_{lb}^t + s \leq x \leq x_{ub}^t - s \right\} \tag{4.29}$$

Remark 4. *The inequality relationship (4.10) is defined in terms of the norm of vectors and matrix. However, in practice, the shrinkage amount s should be a vector that has the same dimension of the state, as the shrinkage required for each state needs to be determined individually. Thus, x_{max}^d is selected to be the accumulated effect of w on x that has the largest norm, which mimics the idea of (4.10), however in the state space.*

Remark 5. *In Algorithm 3, the value of the sensitivity matrix needs to be calculated $2^{n_z+n_w}$ times, as every combination of the maximum and minimum of the state, input, and disturbance are considered. For higher-dimension systems, the computational effort may be significant. One way to reduce the computational cost is to utilize available physical knowledge in the implementation. For example, the optimal operating condition may be estimated based on the nominal model, or it is known to be on one edge of the zone. In these cases, the sensitivity matrix may be calculated only at the estimated optimal operating point, or in the extremes on one edge.*

Furthermore, (4.29) assumes that the target set shrinks in the same amount on the upper and lower bound. If the optimal operating point is known to be on one edge of the target zone, then the shrinkage can be applied to that edge only. This again helps to preserve a larger modified target zone and terminal zone, which helps to reduce the economic performance loss and feasibility issue.

4.5 Consideration of Economic Objectives

A natural extension to the proposed ZMPC (4.7) is to include an economic objective, as the proposed design provides the system with more degrees of freedom to optimize additional objectives. The proposed ZMPC design that handles additional economic objective is

presented as follows:

$$\min_{u_0,\cdots,u_{N-1}} \sum_{i=0}^{N-1} \tilde{\ell}_z(\tilde{x}_i) + \ell_e(\tilde{x}_i, u_i) \quad \text{s.t. (4.7b)} - \text{(4.7f)} \tag{4.30}$$

With weighting factors being large enough on the zone control objective, namely c_1 in (4.8), the controller will prioritize the zone-tracking objective over the economic objective such that the economic objective will be optimized only after the system enters the target zone [41]. Thus the stability properties of the controller (4.30) remain unchanged from that of (4.7). In the following section, simulation results based on (4.30) are presented.

Remark 6. *By tuning the parameter γ in (4.28), a trade-off between the size of the initial feasible region, the economic performance, and zone-tracking performance can be achieved due to changing target zone shrinkage s. A larger γ leads to a smaller modified target zone and correspondingly, a smaller terminal zone. This enhances convergence to the actual target zone while potentially reducing the flexibility and feasibility of the controller, as the system is forced to converge into a smaller region at the end of the control horizon. Higher sacrifice in the economic performance may be observed as well. On the other hand, smaller γ poses less effect on the controller feasibility and economic performance but the closed-loop system has a greater chance to violate the actual target zone.*

4.6 Simulations

In this section, we use a continuous stirred tank reactor (CSTR) as the benchmark process to demonstrate the effectiveness of the proposed design. After introducing the CSTR process (Section 4.6.1) and the detailed setting of the proposed controller (Section 4.6.2), the performance of the proposed controller is first validated without and with a secondary economic objective with respect to the nominal ZMPC in Section 4.6.3. The importance of shrinking

the target set in the proposed controller design is verified in Section 4.6.4. The significance of the terminal constraint in ensuring stability is investigated in Section 4.6.5. Finally, the impact of the tuning parameter γ is investigated in Section 4.6.6.

4.6.1 Process description

The benchmark CSTR process is introduced in this section. An irreversible first-order exothermic reaction $A \rightarrow B$ takes place inside the CSTR reactor, where A and B are the reactants and the desired product, respectively. A cooling jacket is used to control the temperature of the CSTR. Assuming perfect mixing and constant mixture volume inside the reactor, the system can be described by the following ordinary differential equations:

$$\frac{dC_A}{dt} = \frac{q}{V}(C_{A_f} - C_A) - k_0 \exp\Big(-\frac{E}{RT}\Big)C_A \tag{4.31a}$$

$$\frac{dT}{dt} = \frac{q}{V}(T_f - T) + \frac{UA}{V\rho C_p}(T_c - T) + \frac{-\triangle H}{\rho C_p} k_0 \exp\Big(-\frac{E}{RT}\Big)C_A \tag{4.31b}$$

where C_A $[mol/L]$ denotes the molar concentration of the reactant, T $[K]$ represents the temperature inside the reactor, T_c $[K]$ represents the temperature of the cooling stream, and C_{A_f} $[mol/L]$ and T_f $[K]$ denote the molar concentration of the reactant and the temperature of the feed stream, respectively. q $[L/min]$ represents the volumetric flow rate of the streams entering and leaving the reactor, V $[L]$ and ρ $[g/L]$ denote the volume and the density of the mixture inside the reactor respectively, and k_0, E and R are reaction-related parameters, namely the pre-exponential factor of the reaction rate, the activation energy, and the universal gas constant. C_p denotes the specific heat capacity of the mixture, and $\triangle H$ and UA denote the heat of reaction and the heat transfer coefficient between the reactor and the cooling jacket, respectively. The values of the parameters are adopted from [40].

Recall the system defined in (4.1). For the CSTR example specifically, we define the system state vector to be $x = [C_A, T]^T$ and the control input to be $u = T_c$. The disturbance vector is defined as $w = [C_{A_f} - \bar{C}_{A_f}, T_f - \bar{T}_f]^T$, where $\bar{C}_{A_f} = 1.0\, mol/L$ and $\bar{T}_f = 350\, K$ are

the nominal values of the corresponding variables. The following bounds are employed for the state, input, and disturbance, respectively:

$$[0.0, 345.0]^T \leq x \leq [1.0, 355.0]^T \tag{4.32}$$

$$285.0 \leq u \leq 315.0 \tag{4.33}$$

$$[-0.1, -2.0]^T \leq w \leq [0.1, 2.0]^T \tag{4.34}$$

4.6.2 Controller setup and implementation

The proposed ZMPC controller is implemented on the CSTR system both without and with a secondary economic objective. Following the design proposed in (4.7) and (4.30), we define the zone control objective and economic objective. The zone control objective is to hold the reactor temperature T between $348.0\,K$ to $352.0\,K$. Recall the definition of ℓ_z in (4.6), which implies the following:

$$\mathbb{X}_t = \{x : [0.0, 348.0]^T \leq x \leq [1.0, 352.0]^T\}$$

The largest CIS inside the original and modified target zones are approximated using the algorithm proposed in [69].

We computed the value of x^d_{max} to be 0.511 for the states inside the largest CIS with the input and disturbances in their stated boundaries. It was found that the disturbance has the strongest effect on the state at the following point:

$$x = [0.754, 352.0],\ u = 315.0,\ w = [0.1, 2]$$

Unless otherwise mentioned, $\gamma = 1$ is used, which means $s = x^d_{max} = 0.511$. The economic objective is to minimize the concentration of the reactant inside the reactor, i.e. $\ell_e = C_A$. For all the simulations, the control horizon $N = 5$ unless otherwise mentioned.

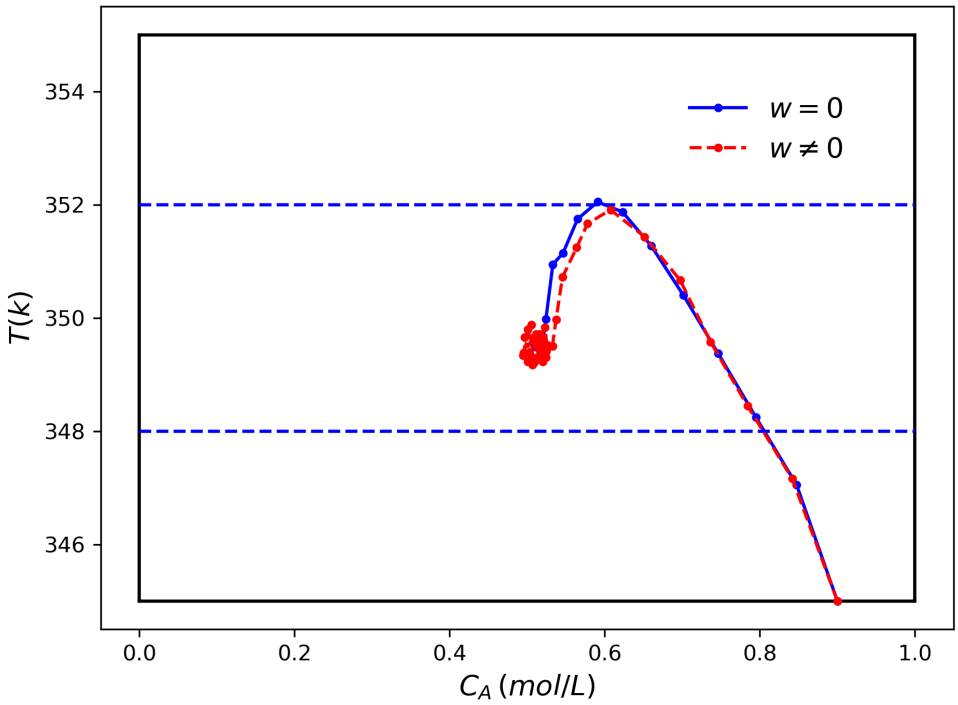

Figure 4.3: Proposed ZMPC without economic objectives.

The value of the disturbance $w(n)$ at any time step is selected randomly inside the boundaries specified by (4.34). Except for the CIS approximation that is carried out in Julia, all other simulations are performed in Python. The optimization problems are solved using the CasADi toolbox developed by [50].

4.6.3 Control performance validation

In this section, we validate the performance of the proposed controller. Figure 4.3 presents the state trajectories under the proposed ZMPC without any economic objective (4.7) with and without the presence of disturbance. Figures 4.4 and 4.5 present the performance of the proposed ZMPC with economic objective (4.30) in comparison to the nominal ZMPC based on different initial conditions. In all figures, the solid black box represents the overall state space, and the dashed blue and red lines denote the boundaries of the original target set $\mathbb{X}_t$ and the modified target set $\tilde{\mathbb{X}}_t$ respectively. The shaded areas represent the largest CIS

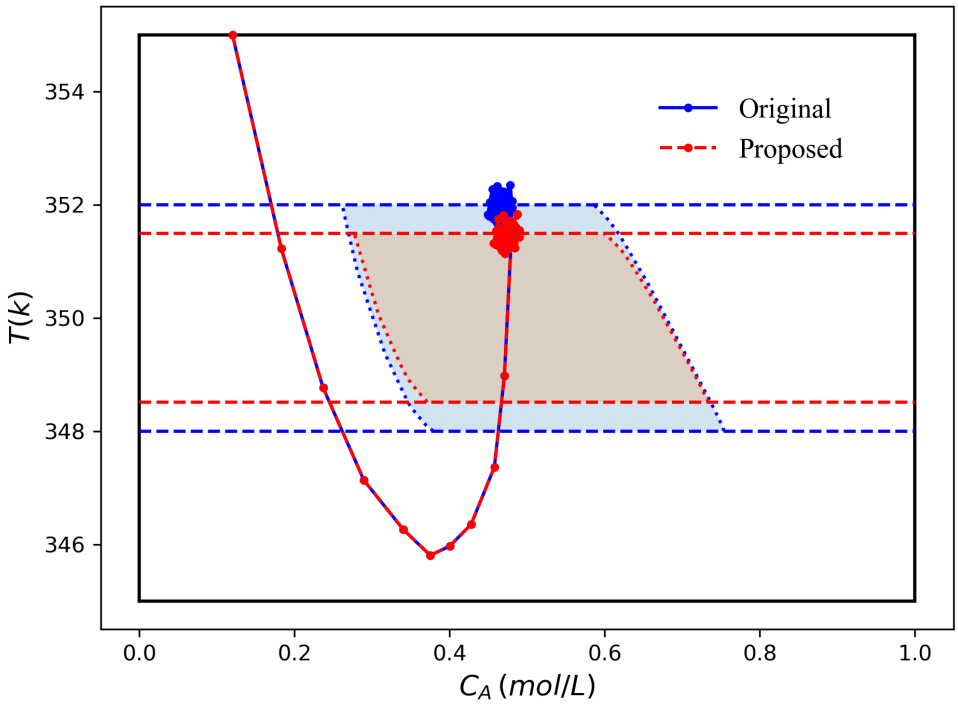

Figure 4.4: Proposed ZMPC v.s. the nominal ZMPC with economic objective, $x_0 = [0.12, 355]$.

inside $\mathbb{X}_t$ and $\tilde{\mathbb{X}}_t$. The larger shaded area represents $\mathbb{X}_t^M$, while the smaller one is $\tilde{\mathbb{X}}_t^M$.

It can be observed from the blue trajectory in Figure 4.3 that without any economic objective, the system converges to a steady-state in the center of the target zone. In the presence of disturbance, as represented by the red trajectory, the state oscillates around the optimal steady-state but is able to stay inside the terminal CIS once entered. Thus, for this particular setup, the benefit of the proposed controller is not significant.

To validate the benefits of the proposed approach, the secondary economic objective is added to the system. With the economic objective considered, the optimal operating condition shifts to the boundary of the target zone. In Figures 4.4 and 4.5, starting from different initial conditions, the state progressions under the control of the nominal ZMPC are shown in blue, while the state trajectories controlled by the proposed ZMPC are red. Under the nominal ZMPC, the state leaves the terminal CIS after converging to the optimal operating point due to the fluctuation caused by the presence of disturbance. These violations

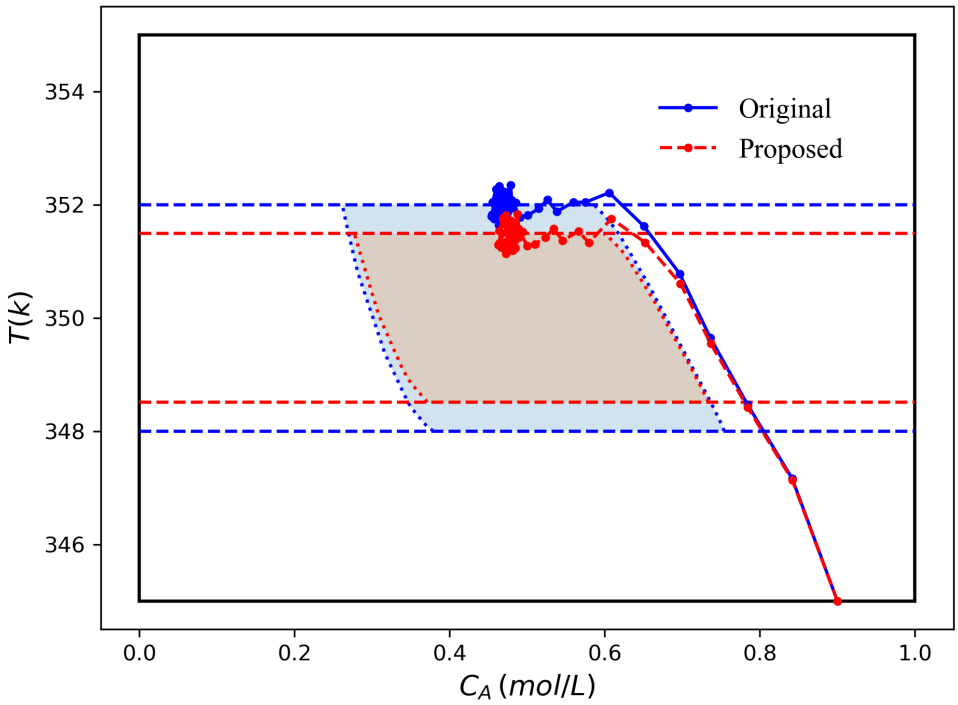

Figure 4.5: Proposed ZMPC v.s. the nominal ZMPC with economic objective, $x_0 = [0.9, 345]$.

on the zone-tracking objective are unfavored and should be eliminated. As presented by the red trajectories, violations with respect to the actual target zone are avoided by the proposed ZMPC. Although the state trajectories violate the upper bound of the modified target zone $\tilde{\mathbb{X}}_t$, they are still enclosed in the actual target zone $\mathbb{X}_t$. Furthermore, Figures 4.4 and 4.5 verify that the proposed controller is able to drive the system state to the same economically-optimal neighborhood even under different initial conditions.

4.6.4 The significance of modifying the target set

One natural question that rises is that instead of modifying just the terminal set, why both the tracking target set and the terminal set are modified in the proposed approach. Figure 4.6 investigates the controller performance if the original target set $\mathbb{X}_t$ is tracked and only the terminal set is modified. The state trajectories under the control of the nominal ZMPC (blue solid trajectory with hexagon markers), the proposed ZMPC (red dashed trajectory

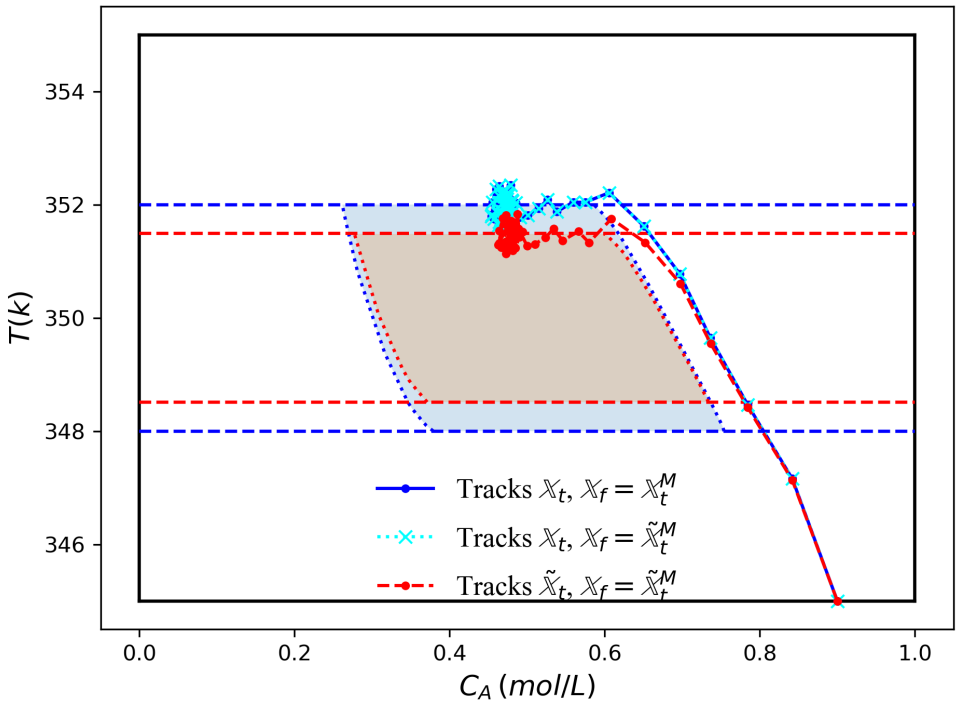

Figure 4.6: ZMPC with economic objective that tracks the original target set ($\mathbb{X}_t$) with modified terminal set ($\mathbb{X}_f = \mathbb{X}_t^M$).

with dot markers), and the ZMPC that tracks the original target set $\mathbb{X}_t$ and the modified terminal set $\tilde{\mathbb{X}}_t^M$ (cyan dotted trajectory with cross markers) are presented, respectively. It can be observed that if only the terminal set is modified, the optimal state trajectory is identical to that obtained based on the nominal ZMPC. This implies that the control performance is not affected if only the terminal set is modified. This matches the theory proposed in [39], which indicates the controller performance remains equivalent in terms of stability as far as $\tilde{\mathbb{X}}_f \subseteq \mathbb{X}_t^M$. Note that the state trajectory obtained based on the proposed ZMPC is presented in the figure for reference, which verifies the effectiveness of modifying the tracking target set.

4.6.5 The significance of the terminal constraint

The significance of the terminal constraint in ensuring closed-loop stability is investigated in this section. With the control horizon $N = 3$, Figure 4.7 shows the state trajectory

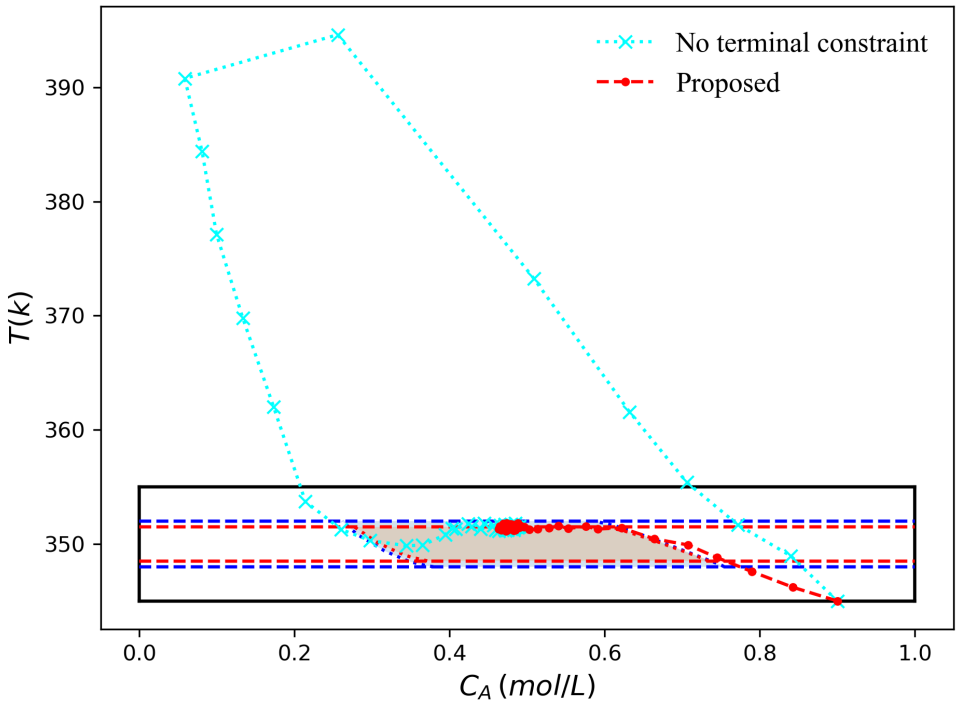

Figure 4.7: ZMPC without the terminal constraint v.s. the proposed ZMPC with economic objective.

under the ZMPC without any terminal constraint in cyan with cross markers, while the trajectory obtained based on the proposed controller is presented in red with dot markers for reference. Without the terminal constraint, the controller is able to drive the system to the optimal operating point, however with aggressive violations of the hard state constraint represented by the solid black box. These violations are avoided by the proposed controller with the terminal constraint. Furthermore, it can be observed that under the controller without terminal constraint, the system takes significantly more steps to converge compared to that taken based on the proposed approach, indicating the effectiveness of the terminal constraint in ensuring closed-loop stability and enhancing convergence.

4.6.6 The impact of γ

The impact of the tuning parameter γ is investigated in this section. Figure 4.8 presents the modified target sets and the corresponding maximum CIS inside the modified target

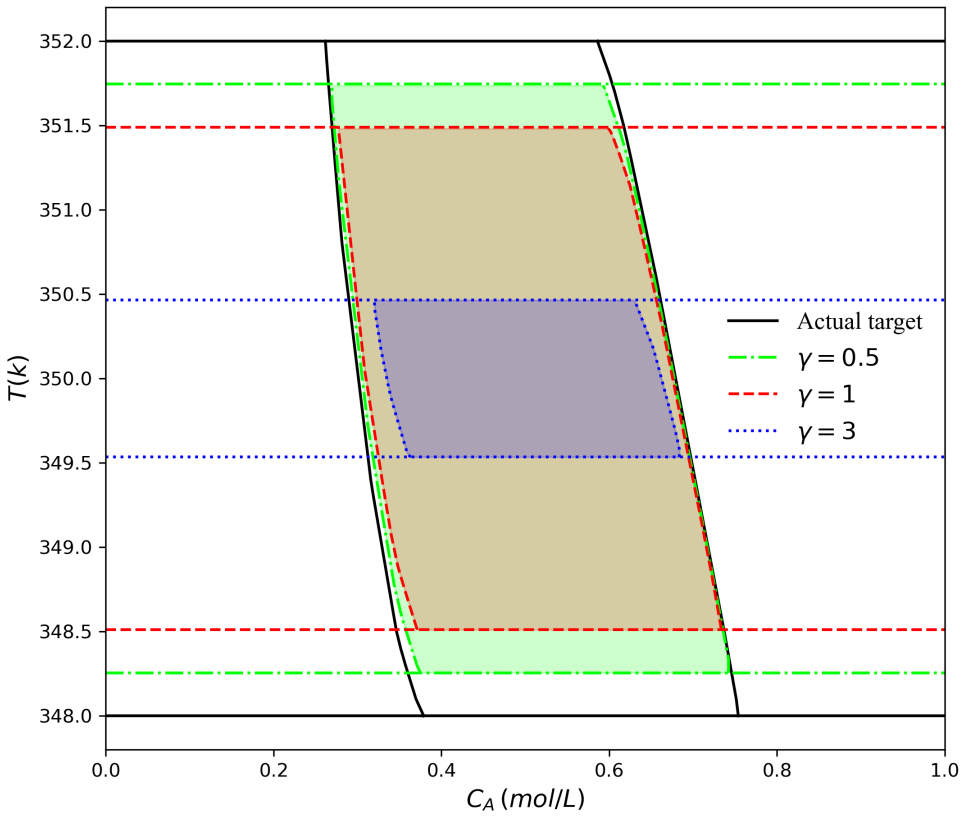

Figure 4.8: The modified target sets and the corresponding maximum CISs under different γ.

sets with different values of γ. Figure 4.9 displays the accumulated zone-tracking cost and the economic cost as γ increases. The accumulated zone-tracking costs are presented in blue with pentagon markers, while the accumulated economic costs are shown in red with circle markers. Note that the accumulated zone-tracking cost is calculated based on the actual target set for a fair comparison, as the modified target zone provided to the controller changes with different γ. Table 4.1 acts as a supplement of Figure 4.9 and presents the number of violations with respect to the actual target set after the system converges to the optimal operating neighborhood for the first time, plus the average magnitude of violation out of all the violations with different γ.

The actual target set is bounded by solid black lines in Figure 4.8. The sets with γ equal to 0.5, 1, and 3 are bounded by green dash-dotted, red dashed, and blue dotted curves,

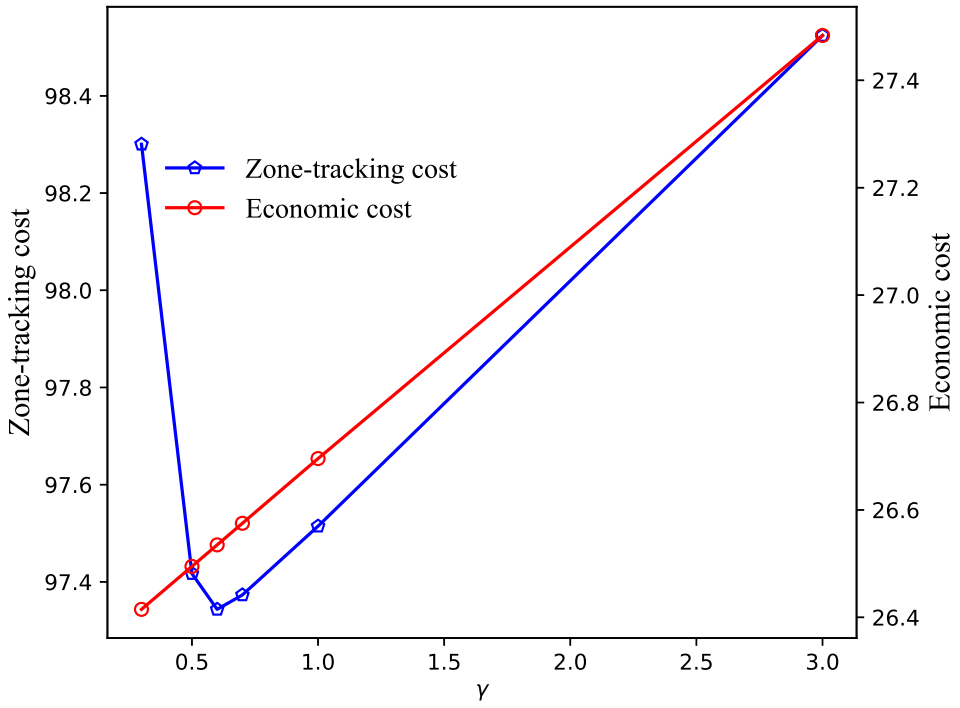

Figure 4.9: The zone-tracking performance and economic performance under different γ.

Table 4.1: The number of violations and the average violation value after the first convergence to the actual target set with different γ.

γ	# of Vio.	Avg. Vio.
0.3	9	0.097
0.5	4	0.045
0.6	2	0.027
0.7	0	-
1	0	-
3	0	-

respectively. The information regarding the optimal operating condition is not considered, both the upper bound and the lower bound of the target zone are shrunk equally. Figure 4.8 indicates the size of the target set and the corresponding largest terminal set available

to the controller reduce as the value of γ increases. This implies the controller becomes more conservative with a smaller feasible region, which leads to a sacrifice in the economic performance. The economic performance is affected in two ways. First, since a smaller feasible zone leads to a shifting in the feasible economically-optimal operating condition. Second, with a more conservative target and terminal zone, the controller tends to drive the system into the zone more aggressively and potentially sacrifice the transit economic performance. This can be seen from Figure 4.9 that the accumulated economic cost function continuously increases as γ increases.

It is however noteworthy that a more conservative controller does not necessarily leads to better zone-tracking performance. It is observed that the accumulated zone-tracking cost first decreases and then increases after reaching a minimum at $\gamma = 0.6$. From Table 4.1, it can be observed that at $\gamma = 0.6$, two violations are observed with a very small magnitude of the average violation. At $\gamma = 0.7$, no violation is observed after convergence, however, the accumulative zone-tracking cost is slightly higher than that obtained with $\gamma = 0.6$. The reason for the increase in zone-tracking cost as γ increase is due to the shrinkage in the feasible operating range and the more aggressive control action chosen by the controller, which sacrifices the transit performance of the system. A larger γ helps to reduce the zone-violation and thus the accumulative tracking cost; however, this reduction is saturated once the violations are insignificant ($\gamma = 0.6$) or are fully eliminated ($\gamma = 0.7$). The sacrifice in the transit performance on the other hand, continuously increases as γ increase, which slowly overtakes the cost reduced by shrinking the target zone.

Remark 7. *It is observed from Figure 4.9 and Table 4.1 that the optimal zone-tracking performance is not achieved at $\gamma = 1$. This indicates that directly shrinking the target zone by the x^d_{max} only provides a conservative estimation of the amount of shrinkage required. It is essential to tune γ to obtain a better zone-tracking performance. On the other hand, it is noteworthy that the economic performance always reduces as γ increases. Thus, it is recommended to use the smallest γ that provides reasonable zone-tracking performance in*

applications.

4.7 Summary

In this chapter, we proposed a robust ZMPC formulation with guaranteed convergence to the actual target set in the presence of bounded disturbance. This is achieved by modifying the actual target set, which helps to reject the effect of the disturbance. A terminal constraint is utilized to ensure the closed-loop stability of the system. The system state is forced to converge to a CIS inside the modified target set at the end of the control horizon. Without assuming the existence of an optimal steady-state operating condition, this generalized approach provides more degrees of freedom. Furthermore, the proposed design is able to handle a secondary economic objective without affecting closed-loop stability. Apart from the theoretical stability proof, an algorithm for determining the modified target set is provided. The proposed design is applied to a CSTR system and has proved to be effective from various perspectives. The effect of the tuning parameter in the proposed algorithm is investigated with the rule of thumb for parameter selection provided.

Chapter 5

Robust ZEMPC Application to an amine-based post-combustion CO_2 capture process

Global warming is a significant global issue that needs to be addressed with high priority. Among various contributing factors, reducing the emission quantity of anthropocentric greenhouse gases, especially CO_2, from industrial processes is an essential way to help address this issue. To be more specific, a major source of CO_2 emission is the electricity generation facilities powered by fossil fuel (e.g. natural gas, coal, etc.) combustion [70]. Although it has been shown that switching the power source to low-carbon energy sources such as renewable energies can reduce the carbon footprint of these processes [71], these energy sources are currently insufficient to meet the continuously increasing demand. Thus, it is essential to investigate approaches that reduce carbon emissions for power plants that utilize fossil fuel while providing time to let the low-carbon energy generation practices mature. Various approaches for CO_2 emission reduction have been proposed. For example, pre-combustion [72], post combustion [73], and oxy-fuel combustion [74] approaches.

One of the state-of-the-art practices is the amine-based post-combustion CO_2 capture

process (PCC), which is the most promising and viable approach available at the moment. Amine-based PCCs can be retrofitted relatively easily into existing power plants. An amine solvent is used to absorb the CO_2 in the post-combustion flue gas generated by the power plant, which helps to reduce the amount of CO_2 released into the atmosphere. However, research has shown that the attachment of an amine-based PCC reduces the power plant efficiency by roughly 10% with the state-of-art methamphetamine (MEA) solvent due to the high energy requirement for solvent regeneration [75]. Maintaining the CO_2 concentration in the solvent within a certain range in the absorption unit is shown to be vital for the efficient operation of the regeneration unit [76]. Thus, the application of advanced process control is essential for sufficient operation of amine-based PCC plants. Various works have been performed in this field, for example, a linear MPC is investigated by Panahi and Skogestad [77], scheduling and MPC have been used collectively in [78], and the operational flexibility of a PCC that is attached to a load-following power plant is studied based on MPC in [79].

In this chapter, we extend the robust ZMPC with the economic objective proposed in the previous chapter [80] and apply it to a large-scale process with nonlinear dynamics, namely the Post-Combustion Carbon Capture (PCC) process. Instead of aiming for the convergence of all states into a terminal set, the constraint is relaxed such that the convergence is required for system outputs only, which helps to improve the feasibility of the proposed controller. Section 5.1 introduces the PCC plant of interest and discusses the first-principle model employed, which is developed by [81]. The control objective is defined in Section 5.2. The proposed algorithm for the system-output-based terminal set computation is introduced in Section 5.3. Section 5.4 presents the simulation performance of the proposed controller. The proposed controller is compared with multiple ZMPC controller designs and is shown to be beneficial in terms of disturbance rejection and stability.

5.1 Preliminaries

5.1.1 Process Overview

A schematic diagram of the PCC process of interest is presented in Figure 5.1. The process consists of two major operating units, specifically the absorption and desorption columns. Flue gas with high CO_2 concentration enters the process from the bottom of the adsorption column and is in contact with the lean solvent (low in CO_2 concentration) that enters the column from the top. Reaction-based separations take place in the column and the treated flue gas leaves the column at the top with low CO_2 concentration. The rich solvent with high CO_2 concentration exits the column at the bottom and is sent off to the desorption column upon passing through a heat exchanger, where the rich solvent is heated by the lean solvent exiting the reboiler. CO_2 is separated from the lean solvent in the desorption column and leaves the column at the top as the product gas. The concentration of CO_2 in the product gas is typically between 90% and 99%. The solvent flows through the desorption column and reaches the reboiler, where it is heated by steam to about 120°C. Part of the liquid solvent vaporizes in the reboiler and moves up the desorption column, while the rest is recycled back to the absorption column as the lean solvent upon cooled down to the appropriate temperature. Make-up solvent is added to the absorption column as needed to compensate for the loss of solvent throughout the process.

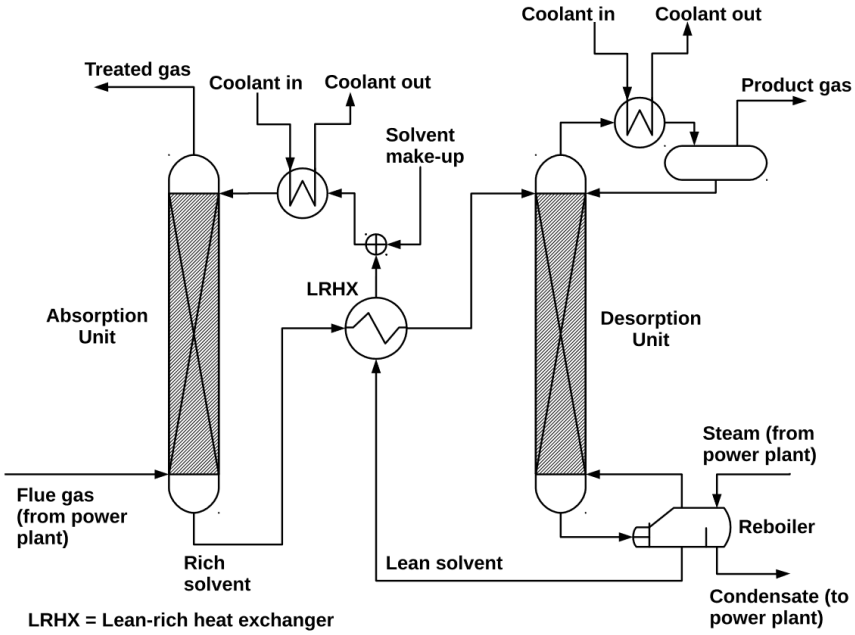

Figure 5.1: A schematic diagram of an amine-based post-combustion carbon capture plant (PCC).

5.1.2 Model Description and Control Objective

The first principle model developed by [81] is employed in this chapter to describe the system dynamics. The model is a system of Differential Algebraic Equations (DAEs) that can be represented in the following form:

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

$$g(x, z, u, w) = 0 \tag{5.1b}$$

$$y = h(x, z, u, w) \tag{5.1c}$$

where $x \in \mathbb{R}^{103}$ and $z \in \mathbb{R}^{7}$ represent the differential and algebraic state variables, respectively. $u \in \mathbb{R}^{2}$ denotes the manipulated input, $w \in \mathbb{R}$ represents the system disturbance. y is the system output. $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are nonlinear functions.

For the completeness of this chapter, we briefly discuss the physical meaning of the

process variables introduced in (5.1). More details on the model can be found in [81]. Four components are considered in this model, namely CO_2, MEA, H_2O, and N_2. Each column is discretized into 5 sections along the vertical spatial axis, with the dynamics of each section represented by the dynamics at the center of them. The differential states are the composition of each component and the temperature at each column section, plus the temperature of the shell side and the tube side of the heat exchanger and the temperature inside the reboiler. The algebraic states are the liquid composition of each component in the reboiler, the reboiler vapor fraction, the CO_2 concentration in the liquid stream going into the absorption column, and the gas flow rate entering the desorption column, respectively. Two system outputs are considered, namely the percentage of CO_2 captured in the absorption column, and the reboiler temperature T_{reb}, which is to be monitored closely kept in a relatively tight range to ensure the smooth operation of the entire process. The percentage of CO_2 captured is defined as follows:

$$\text{CR}_{\text{CO}_2} = \frac{\text{Molar flow rate of CO}_2\text{ in} - \text{Molar flow rate of CO}_2\text{ out}}{\text{Molar flow rate of CO}_2\text{ in}} \times 100\%$$

The manipulated inputs of the process are the lean solvent flow rate F_{MEA} and the energy supply rate to the reboiler Q_{reb}. The system disturbance is the fluctuation of the flue gas flow rate entering the PCC plant with respect to its nominal value. It is assumed that at any given time, the actual magnitude of w is unknown but bounded and has the origin in its domain, such that the following relationship is satisfied:

$$w \in \mathbb{W} := \{w \in \mathbb{R} : ||w||_\infty \leq \theta, \theta > 0\} \tag{5.2}$$

where the infinity norm of w is bounded by a positive constant θ.

5.2 Robust Economic Model Predictive Control with zone-tracking

The primary control objective for the PCC plant is to optimize the CO_2 treatment performance, in other words, maximize CR_{CO_2} while keeping the lean solvent consumption and the energy consumption in the reboiler reasonable. To achieve these control objectives in the presence of system disturbance, we propose a robust EMPC with zone-tracking. The zone-tracking objective aims to maintain the system outputs in a certain range, and the economic objective penalizes the solvent and energy consumption. The proposed controller is discussed in detail in the following section.

We extend the robust EMPC with zone-tracking proposed in Chapter 4 and apply it to the PCC process. The terminal constraint is modified such that only the system outputs, instead of the system states, are tracked, which improves the feasibility of the controller on large-scale systems.

At time instant t_k, the proposed robust ZMPC with economic objective can be represented as follows:

$$V_N^0(x(t_k)) = \min_u \int_{t_k}^{t_{k+N}} \ell_z(\tilde{y}(t)) + \ell_e(\tilde{x}(t), z(t), u(t)) \tag{5.3a}$$

$$\text{s.t. } \dot{\tilde{x}}(t) = f(\tilde{x}(t), \tilde{z}(t), u(t), 0) \tag{5.3b}$$

$$g(\tilde{x}(t), \tilde{z}(t), u(t), 0) = 0 \tag{5.3c}$$

$$\tilde{y}(t) = h(\tilde{x}(t), \tilde{z}(t), u(t), 0) \tag{5.3d}$$

$$\tilde{x}(t_k) = x(t_k) \tag{5.3e}$$

$$\tilde{x}(t) \in \mathbb{X} \tag{5.3f}$$

$$\tilde{z}(t) \in \mathbb{Z} \tag{5.3g}$$

$$u(t) \in \mathbb{U} \tag{5.3h}$$

$$\tilde{y}(t_{k+N}) \in \tilde{\mathbb{Y}}_f \tag{5.3i}$$

where similar notations are used as those used in (4.7).

The objective function (5.3a) consists of two parts, namely the zone-tracking objective $\tilde{\ell}_z(\cdot)$ and the economic objective $\tilde{\ell}_e(\cdot)$. The economic objective captures the heat duty of the reboiler and is defined as follows:

$$\ell_e(x, z, u) = \frac{Q_{\text{reb}}}{F_{\text{MEA}} \cdot RV_{\text{reb}} * M_{\text{CO}_2}} \tag{5.4}$$

where F_{MEA} and Q_{reb} are the system input, RV_{reb} is the vapor fraction in the reboiler, M_{CO_2} is the molar mass of CO_2.

$\tilde{\mathbb{Y}}_t$ represents the zone-tracking set seen by the controller. It is worth mentioning that $\tilde{\mathbb{Y}}_t$ is a subset of the actual target set $\mathbb{Y}_t$, which is defined based on the first principle knowledge regarding the ideal system operating condition.

Similar to that discussed in Chapter 4, we define the actual target set $\mathbb{Y}_t$, and the modified target set $\tilde{\mathbb{Y}}_t$, which represents the target set defined based on first-principle knowledge and the target set seen by the controller, respectively. The modification in the target set helps provide a buffer zone for disturbance rejection and thus improves the robustness of the proposed controller. The following relationship between the modified target set, the actual target set, and the output space $\mathbb{Y}$ holds:

$$\tilde{\mathbb{Y}}_t \subset \mathbb{Y}_t \subset \mathbb{Y}$$

where the output space $\mathbb{Y}$ is defined as follows:

$$\mathbb{Y} := \{y = h(x, z, u, p) | x \in \mathbb{X}, z \in \mathbb{Z}, u \in \mathbb{U}, p \in \mathbb{P}\}$$

Note that instead of aiming for the convergence of all the states into a control invariant

terminal set as proposed in Chapter 4, in this chapter, we relax the terminal constraint and only focus on the system output. This helps enhance the feasibility of the proposed controller on large-scale systems while still keeping the closed-loop system robust. In the following section, we review the detailed procedure of determining the terminal set $\mathbb{Y}_f \in \tilde{\mathbb{Y}}_t$.

5.3 Determining the Control Invariant Terminal Set

In this section, we present the approach taken to obtain the control invariant terminal set. Section 5.3.1 introduces the approach taken for approximating the CIS in the state space, while Section 5.3.2 presents the method used to project the CIS into the output space.

5.3.1 Approximation of CIS in the state space

The proposed algorithm for obtaining the terminal set $\mathbb{Y}_f$ is an extension of that proposed in [82]. The algorithm is built based on the algorithm developed by Polyak and Shcherbakov [83] that compute an ellipsoidal inner approximation of the CIS for linear systems in the form of:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u} \tag{5.5}$$

where $\bar{x} = x - x_s$ and $\bar{u} = u - u_s$ are the deviation form of the state and input with respect to the steady-state nominal operating point (x_s, u_s). The ellipsoidal CIS is obtained by solving the following semi-definite program (SDP):

$$\max_{P,Y} \quad \text{trace}(P) \tag{5.6a}$$

$$s.t. \quad AP + PA^T + BY + Y^T B^T \prec 0 \tag{5.6b}$$

$$\begin{bmatrix} P & Y^T \\ Y & \bar{u}_{\max}^2 I \end{bmatrix} \succeq 0 \tag{5.6c}$$

where P and Y are matrices with appropriate dimensions, $\bar{u}_{\max} \leq ||u||$, $\forall u \in \mathbb{U}$ upper bounds the 2-norm of the input vector. The matrix P is assumed to be positive definite and symmetric (i.e. $P^T = P$, $P \succ 0$). The maximum ellipsoidal inner approximation of the CIS can be represented based on the matrix P as follows:

$$\mathbb{X}_I := \{x | x \in \mathbb{X}, \bar{x}^T P \bar{x} \leq 1\} \tag{5.7}$$

Note that if the linear system (5.5) is stable, the maximum ellipsoidal CIS in the state space can be calculated by solving the continuous Lyapunov equation $AP + PA^T + Q = 0$. In this case, the maximum ellipsoidal CIS can be described as follows:

$$\mathbb{X}_I := \{x | x \in \mathbb{X}, \bar{x}^T P \bar{x} \leq \alpha\} \tag{5.8}$$

where α is a tuning parameter that ensures $\mathbb{X}_I \in \mathbb{X}$. To implement the algorithm introduced above, the nonlinear system of interest needs to be linearized first around a steady-state of the system (x_s, u_s). Note that here the algebraic states are excluded from the computation for simplicity, as their effects on the overall system dynamics are relatively minor. The linearized system takes the form of (5.5) with $A = \frac{\partial f}{\partial x}\big|_{x_s,u_s}$, $B = \frac{\partial f}{\partial u}\big|_{x_s,u_s}$.

Note that nonlinear systems have multiple steady-states, and the objective is to obtain the largest $\mathbb{Y}_f \subset \tilde{\mathbb{Y}}_t$. Thus, an iterative algorithm that tunes the steady-state (x_s, u_s) and α until a set that satisfies the objective is developed. In order to tune the steady-state (x_s, u_s) that acts as the center of the linearized model, we first find the optimal economic cost ℓ_e^* that exists in the modified target set by solving the following steady-state optimization problem:

$$\ell_e^* = \min_{x,z,u} \ \ell_e \tag{5.9a}$$

$$s.t.\ f(x, z, u, 0) = 0 \tag{5.9b}$$

$$g(x, z, u, 0) = 0 \tag{5.9c}$$

$$y = h(x, z, u, 0) \tag{5.9d}$$

$$x \in \mathbb{X} \tag{5.9e}$$

$$z \in \mathbb{Z} \tag{5.9f}$$

$$u \in \mathbb{U} \tag{5.9g}$$

$$y \in \tilde{\mathbb{Y}}_t \tag{5.9h}$$

Oftentimes, the state-input pair that provides the optimal economic cost ℓ_e is on the boundary of the set $\tilde{\mathbb{Y}}_t$. To find a different steady-state inside $\tilde{\mathbb{Y}}_t$, the following optimization problem is solved:

$$(x^*, u^*) = \arg\min_{x,z,u} \ 0 \tag{5.10a}$$

$$s.t. \ f(x, z, u, 0) = 0 \tag{5.10b}$$

$$g(x, z, u, 0) = 0 \tag{5.10c}$$

$$y = h(x, z, u, 0) \tag{5.10d}$$

$$\ell_e(x, z, u) = \ell_e^r \tag{5.10e}$$

$$x \in \mathbb{X} \tag{5.10f}$$

$$z \in \mathbb{Z} \tag{5.10g}$$

$$u \in \mathbb{U} \tag{5.10h}$$

$$y \in \tilde{\mathbb{Y}}_t \tag{5.10i}$$

where ℓ_e^r is an economic cost that is worse than the optimal economic cost ℓ_e^* such that:

$$\ell_e^r = (1 + r) \cdot \ell_e^*, \ r \in (0, 1)$$

To tune α, we introduce the tuning parameter $\delta < 1$ to tune down α when needed. Thus, the problem of finding the largest ellipsoidal CIS approximation inside the modified target

set is transferred into finding the best combination of ℓ_e^r and α. Algorithm 4 summarizes the proposed iterative algorithm that returns the CIS approximation $\mathbb{X}_I$ and $\mathbb{Y}_f$ in $N_{\max}$ number of iterations at the maximum.

Algorithm 4 Estimation of the terminal CIS

Input: f, h, ℓ_e, ℓ_e^*, $\mathbb{X}$, $\mathbb{U}$, $\tilde{\mathbb{Y}}_t$, $u_{\max}$, ε, r, α, δ, $N_{\max}$
Output: $\mathbb{X}_I$, $\mathbb{Y}_f$
Ensure: $x \in \mathbb{X}$, $u \in \mathbb{U}$
  $\bar{y}_t = \{max\,(y) \,|\, y \in \tilde{\mathbb{Y}}_t\}$, $\underline{y}_t = \{min\,(y) \,|\, y \in \tilde{\mathbb{Y}}_t\}$
  $\mathbb{X}_0 \leftarrow \mathbb{X}$, $\mathbb{Y}_0 \leftarrow \tilde{\mathbb{Y}}_t$, $\alpha \leftarrow \alpha$
  $i \leftarrow 0$
  while $\mathbb{Y}_i + \varepsilon\mathbb{B} \cap \tilde{\mathbb{Y}}_t \backslash \mathbb{Y}$ **do**
    $i \leftarrow i + 1$
    $\ell_e^r \leftarrow (1 + r)\ell_e^r$
    Define $\bar{y}_i = \{max\,(y) \,|\, y \in \mathbb{Y}_i\}$, $\underline{y}_i = \{min\,(y) \,|\, y \in \mathbb{Y}_i\}$
    if $\bar{y}_i - \underline{y}_i > \bar{y}_t - \underline{y}_t$ **then**
      $r \leftarrow 0$, $\alpha \leftarrow \delta\alpha$
    else$r \leftarrow r$, $\alpha \leftarrow \alpha$
    end if
    Solve the steady-state optimization and obtain (x_s, u_s)
    Linearize the system w.r.t. (x_s, u_s) and obtain A and B
    if A is stable **then**
      Solve the continuous Lyapunov Equation to obtain P
      $\mathbb{X}_i \leftarrow x \in \mathbb{X} | x^T P x \leq \alpha$
      $\mathbb{Y}_i \leftarrow h(\mathbb{X}_i)$
    else
      Solve the SDP to obtain P
      $\mathbb{X}_i \leftarrow x \in \mathbb{X} | x^T P x \leq 1$
      $\mathbb{Y}_i \leftarrow h(\mathbb{X}_i)$
    end if
  end while
  if $i = N_{max}$ **then** $\mathbb{X}_i \leftarrow \emptyset$, $\mathbb{Y}_i \leftarrow \emptyset$
    Break
  end if
  $\mathbb{X}_I \leftarrow \mathbb{X}_i$, $\mathbb{Y}_f \leftarrow \mathbb{Y}_i$
  return $\mathbb{X}_I$

To assist the understanding of Algorithm 4, Figure 5.2 presents a diagram of three potential iterations while Algorithm 4 runs. The solid rectangles represent the modified target set $\tilde{\mathbb{Y}}_t$, the solid ellipses represent the CIS approximation $\mathbb{Y}_i$, and the dashed ellipses are the

relaxed CIS approximation referred to as $\mathbb{Y}_i + \varepsilon\mathbb{B}$ in Algorithm 4, where $\mathbb{B}$ is a unit ball of the appropriate dimension centered at the origin. As presented in Algorithm 4, the algorithm runs until the relaxation $\mathbb{Y}_i + \varepsilon\mathbb{B}$ is fully inside the modified target zone $\tilde{\mathbb{Y}}_t$, as shown in Figure 5.2(3). This means that we compute a relatively conservative approximation of the CIS since the ellipsoid is an approximation obtained based on a linearized model with limited accuracy as the system moves away from the center steady-state (x_s, u_s). As can be seen from Figure 5.2, the shift in the steady-state (x_s, u_s) shifts the center of the obtained CIS, while the change in α changes the size of the CIS. If the steady-state (x_s, u_s) is too close to the boundary of $\tilde{\mathbb{Y}}_t$, as shown in Figure 5.2(1), increasing ℓ_e^r is desired. On the other hand, for the case in Figure 5.2(2), both the maximum and the minimum value of the relaxed CIS $\mathbb{Y}_f + \varepsilon\mathbb{B}$ falls outside the modified target set $\tilde{\mathbb{Y}}_t$, meaning that α is too big and needs to be tuned down.

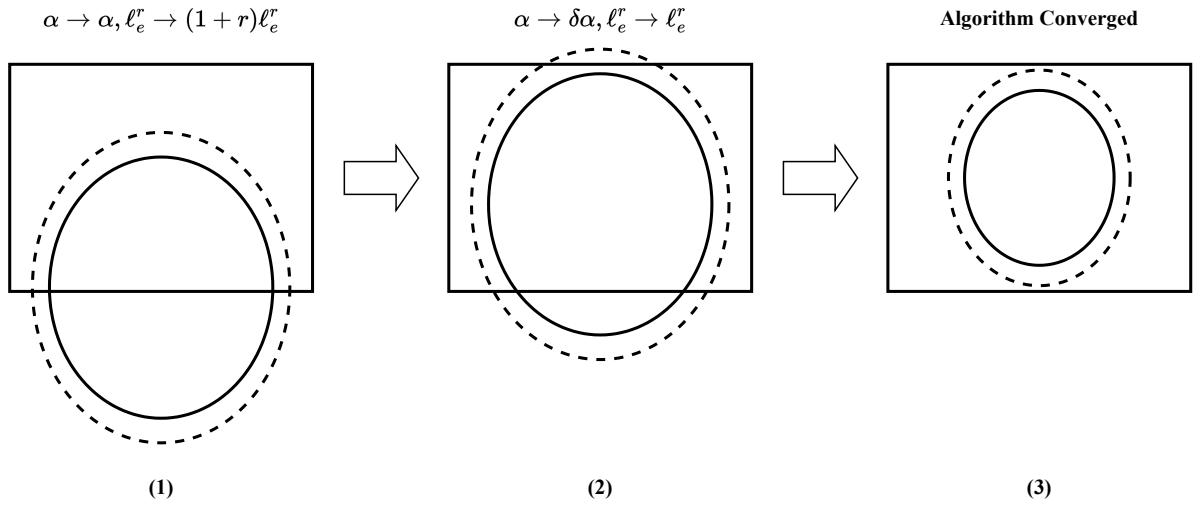

Figure 5.2: An illustrative diagram showing three potential iterations Algorithm 4

5.3.2 Projecting $\mathbb{X}_I$ into the output space

Since the proposed controller enforces the terminal constraint on the system output, we define $\mathbb{Y}_f$ to be the projection of $\mathbb{X}_I$ onto the output space:

$$\mathbb{Y}_f := \{y | y = h(x, z, u), x \in \mathbb{X}_I, z \in \mathbb{Z}, u \in \mathbb{U}\} \tag{5.11}$$

Despite the simplicity of the above definition, it is not in a mathematically explicit form and thus cannot be used directly in (5.3). Taking the geometrical property of the ellipsoidal CIS $\mathbb{X}_I$ into consideration, a method that computes the explicit expression of $\mathbb{Y}_f$ is developed.

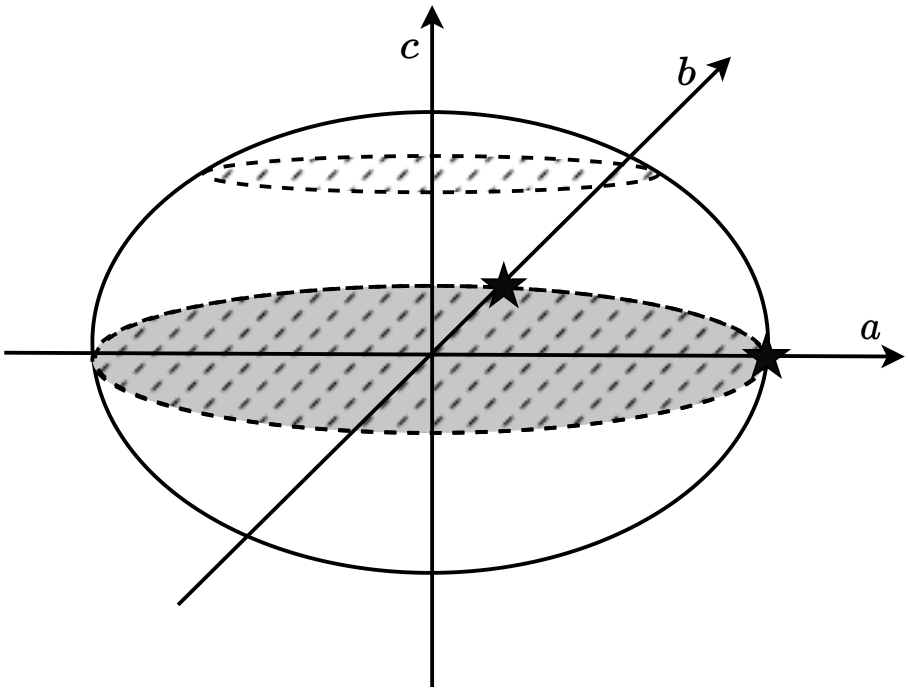

Figure 5.3: A schematic diagram of the relationship between $\mathbb{Y}_f$ and $\mathbb{X}_I$

To enhance understanding of the proposed algorithm, we visualize the relationship between $\mathbb{Y}_f$ and $\mathbb{X}_I$ in a 3-dimensional space, which is shown in Figure 5.3. The system state for this example is $x = [a, b, c]$, with the system output being $y = [a, b]$. The large ellipsoid centered at the origin represents $\mathbb{X}_I$ defined based on the deviation state $\bar{x} = x - x_s$, where

x_s is the steady-state center of the linearized model. It can be observed that the largest cross-section of $\mathbb{X}_I$ in the output space, i.e. $\mathbb{Y}_t$, is obtained when the value of the other system state c is set to 0, which is represented by the shaded ellipse.

Since the set $\mathbb{Y}_f$ is an ellipsoidal shape centered at the origin in the deviation output space, it can be expressed explicitly as follows:

$$\mathbb{Y}_f = \{y \mid y \in \tilde{\mathbb{Y}}_t, \bar{y}^T Q \bar{y} \leq 1\} \tag{5.12}$$

where $\bar{y} = y - y_s$, $y_s = h(x_s, u_s)$, Q is a positive definite symmetric matrix of the correct dimension. To calculate Q, one simple way is to obtain the coordinates of the endpoint on each system output axis, which are highlighted in Figure 5.3 by star markers. Recall that the points on the boundary of $\mathbb{X}_I$ satisfy the following equation:

$$x^T P x = \alpha \tag{5.13}$$

The coordinates of the points marked by star markers can thus be found by setting the states on all other axes to 0. For example, solving the following equation provides us with the boundary point on the a-axis:

$$x_a^T P x_a = \alpha, x_a^T = [a^*, 0, 0] \tag{5.14}$$

With the endpoints for each system output variable calculated, the matrix Q can be solved accordingly. Thus, the explicit form of $\mathbb{Y}_f$ is obtained.

Extending the observation from Figure 5.3, we can calculate $\mathbb{Y}_f$ based on $\mathbb{X}_I$ for the PCC process following a similar procedure. It is noteworthy that the system output of the PCC process is a linear function of the system state. Thus, calculating $\mathbb{Y}_f$ and obtaining the largest hyperplane in the space formed by the output-determining state variables inside $\mathbb{X}_I$ are merely the same problem.

5.4 Simulations

In this section, we validate the performance of the proposed ZEMPC controller in the presence of system noise and disturbance. A ZEMPC that tracks the actual target set without using any terminal constraint is used as the benchmark ZEMPC for comparison and is referred to as the nominal ZEMPC hereafter. Section 5.4.1 introduces the detailed controller setup, while Section 5.4.2 presents the simulation results of the proposed ZEMPC and the nominal ZEMPC along with discussions.

5.4.1 Controller setup and implementation

Recall the system output vector selected for the PCC process $y = [\mathrm{CR}_{\mathrm{CO}_2}, T_{\mathrm{reb}}]^T$, where the two elements are the CO_2 capture efficiency and the reboiler temperature, respectively. The actual target set $\mathbb{Y}_t$ is defined with respect to the two output variables as follows:

$$\mathbb{Y}_t = \{y \mid [90\%, 390]^T \leq y \leq [95\%, 391]^T\} \tag{5.15}$$

Different from the approach taken in Chapter 4, we obtain the ellipsoidal terminal set $\tilde{\mathbb{Y}}_f$ first, and then define the modified target set to be the smallest polyhedron around $\tilde{\mathbb{Y}}_f$, where the smallest polyhedron set around a known set is defined in Definition 4. We used $r = 0.01$, $\alpha = 0.01$, $\varepsilon = 0.008$ for the computation of $\tilde{\mathbb{Y}}_f$. Figure 5.4 shows the spacial relationship of $\mathbb{Y}_t$, $\tilde{\mathbb{Y}}_t$, and $\tilde{\mathbb{Y}}_f$, which are represented by the solid rectangle, the dashed rectangle, and the dashed-dotted ellipse. Note that the carbon capture efficiency is the limiting variable in the terminal set computation, which causes the modified target set and the terminal set to be rather conservative with respect to the boiler temperature.

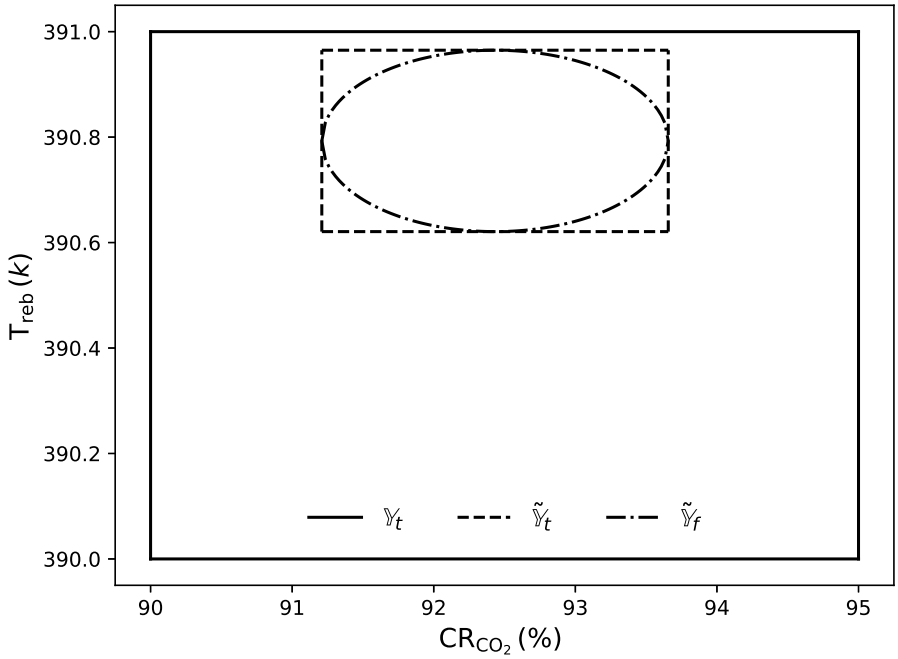

Figure 5.4: The actual target set, modified target set, and the terminal set of the proposed ZEMPC

For fair comparisons, the prediction horizon of the proposed ZEMPC and the nominal ZEMPC are both set to $N = 18$, with a 10 minute sampling interval. The overall simulation time is 15 hours. 2% process noise is considered for both controller designs. Furthermore, the inlet flow rate of the flue gas acts as a system disturbance for both controllers by changing over time around its nominal value, with the deviation from the nominal value upper bounded by 2%.

Remark 8. *The two system outputs are carefully selected such that when they are tightly controlled, the system stability and efficient performance monitoring are both ensured. To be specific, CR_{CO_2} is the key performance measurement to monitor, while T_{reb} is critical for the stable operation of the system. The significance of regulating T_{reb} in terms of system stability can be explained by the first principle. Maintaining a valid phase equilibrium inside the desorption column is necessary for proper system operation. Assuming a realistic stable reboiler pressure, T_{reb} has to be tightly controlled to achieve this.*

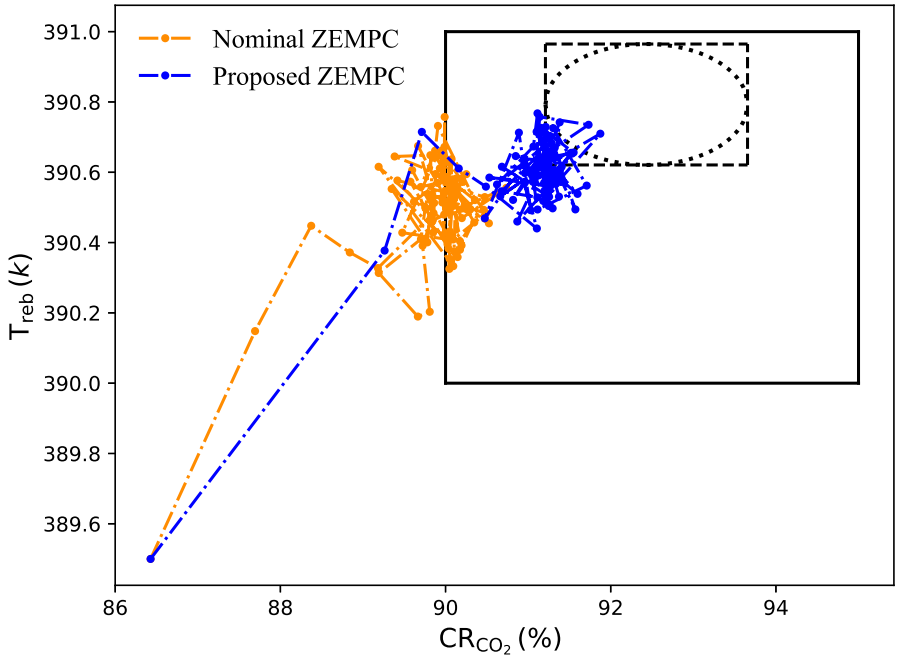

Figure 5.5: The system output trajectories obtained based on the proposed robust ZEMPC v.s. the nominal ZEMPC

5.4.2 Result and discussion

The simulation results of the proposed robust ZEMPC and the nominal ZEMPC are presented in this section. Figure 5.5 presents the output progression in the output space over time, where the orange trajectory is obtained based on the nominal ZEMPC, and the blue trajectory is obtained based on the proposed controller. Figures 5.6 and 5.7 show the zone-tracking cost and the economic cost obtained based on the two controllers over the simulation horizon, respectively. The same colors as those used in Figure 5.5 are used in Figures 5.6 and 5.7. Note that Figures 5.6 and 5.7 focus on presenting how the controllers handle noise and disturbance after the system output has converged to the neighborhood of a steady-state, thus only the costs after $t = 100$ min are shown. The zone-tracking costs are computed with respect to the actual target set for both controllers. Table 5.1 summarizes the total zone-tracking cost and the economic cost obtained based on the two controllers.

Figure 5.5 indicates that the proposed ZEMPC can handle noise and system disturbance

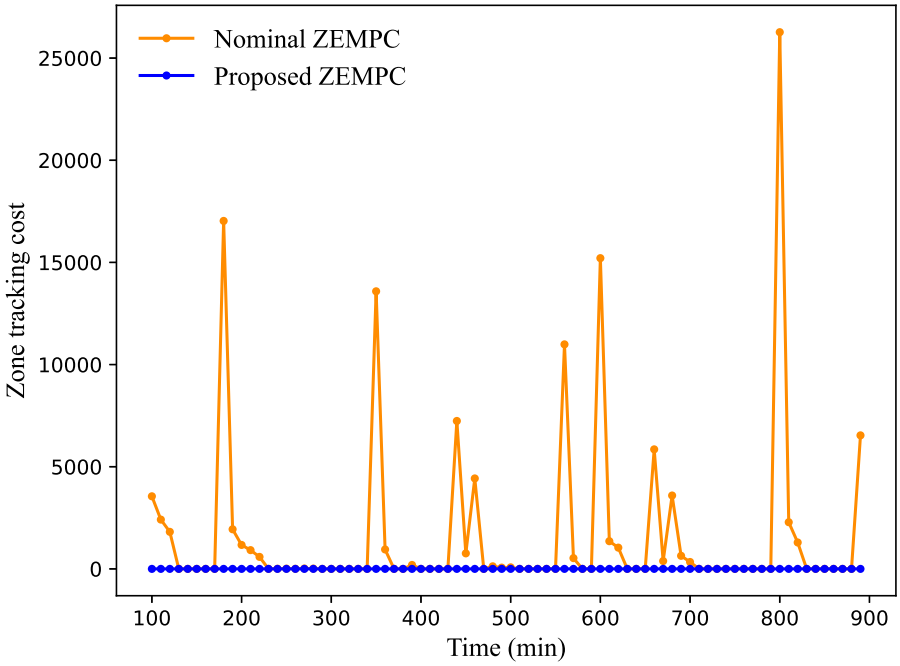

Figure 5.6: The zone-tracking cost over the simulation time obtained based on the proposed robust ZEMPC v.s. the nominal ZEMPC

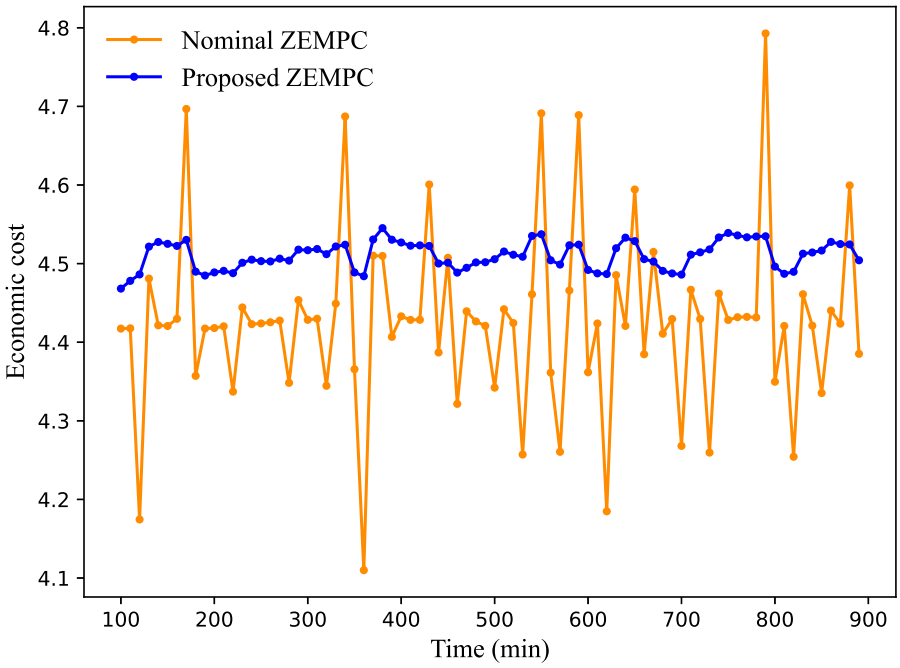

Figure 5.7: The economic cost over the simulation time obtained based on the proposed robust ZEMPC v.s. the nominal ZEMPC

Table 5.1: The total zone-tracking cost and economic cost of the proposed robust ZEMPC v.s. the nominal ZEMPC

	The proposed ZEMPC	The nominal ZEMPC
ℓ_z	5.45×10^5	1.09×10^5
ℓ_e	405	398

better compared to the nominal ZEMPC since it is able to keep the system output inside the actual target set once converged. On the other hand, under the control of the nominal ZEMPC, the system output fluctuates around the boundary of the actual target set and thus violates the target set more frequently. This can also be observed in Figure 5.6, where the proposed ZEMPC leads to no violation of the actual target zone, while the nominal ZEMPC suffers from zone violation over the entire simulation horizon. It is to be noted that the proposed robust ZEMPC enhances the closed-loop robustness with some scarification in the economic performance. This can be observed in Figure 5.7, where the economic cost obtained based on the proposed controller has a larger average value compared to that obtained with the nominal ZEMPC. However, it is also noticed that the proposed ZEMPC leads to economic costs with less variation, which is more robust compared to the nominal ZEMPC. Table 5.1 shows matching results, where the proposed ZEMPC significantly reduces the zone-tracking cost at the price of a slightly higher economic cost compared to the nominal ZEMPC.

Remark 9. *Note that overall, the zone-tracking cost has a much larger magnitude compared to the economic cost. This is designed on purpose in both controllers to prioritize the zone-tracking objective over the economic objective, since the zone-tracking objective is more vital for the safe and robust operation of the process.*

5.5 Summary

In this chapter, we extended the robust ZEMPC proposed in Chapter 4 and applied it to the PCC process, which is a large-scale nonlinear system with complex dynamics. A practical approach for approximating a control invariant set with respect to the system output is

developed, which improves the applicability of the robust ZEMPC. The performance of the proposed controller is compared with the performance of a benchmark nominal ZEMPC in the presence of system noise and disturbance, and is shown to have better capacity for plant-model-mismatch rejection and is more robust.

Chapter 6

Conclusions and Future work

6.1 Conclusions

In this thesis, we systematically investigated advanced-control-oriented model approximation methods and controller robustness in the presence of plant-model-mismatch. We also extended our studies and tested the performance of the proposed methodologies on essential complex real-world applications.

In Chapter 2 we investigated three model approximation approaches, namely the proper orthogonal decomposition (POD) method integrated with trajectory piecewise linearization, the subspace model identification method, and neural network training method with explicit expressions. Each method is applied to two benchmark systems, namely an alkylation process and a wastewater treatment process (WWTP). The performance of the model approximation methods is evaluated in open-loop prediction and closed-loop advanced process control performance. It is found that regardless of the model approximation method employed, a trade-off between the model complexity and model performance needs to be made. Furthermore, it is found that offset-free steady-state prediction is essential for offset-free conventional tracking MPC, while for EMPC, high accuracy in transitional dynamic prediction is vital. NN is found to be a powerful generalized tool for model approximation due to its

ability to predict nonlinear dynamics. However, it is noticed that the NN training process quickly becomes very challenging as the number of system inputs and outputs increases.

In Chapter 3, we continued our study of advanced-control-oriented model approximation for an agro-hydrological system. We selected NN training with explicit expressions as our base and proposed a two-layer NN modeling framework for the system of interest with its unique dynamics taken into consideration. The structure of the proposed framework is inspired by the design of linear time-varying (LPV) models. The operating range of interest is divided into three overlapping smaller operating regions with an NN sub-model developed for each operating region. The three sub-models form the first layer of the proposed framework. The second layer of the proposed framework consists of one fully connected NN, which aims to capture the nonlinear mapping between the predictions made by the sub-models to the actual system output. The proposed framework is tested for ZMPC application and shows better performance compared to a single LSTM NN model. The control performance of ZMPC under different hyperparameters is as well investigated in the presence of system disturbance, where more conservative ZMPC designs are shown to have less zone violations (i.e. more robust closed-loop control performance) with some sacrifice in economic performance.

Followed by the simulation-based investigation of ZMPC robustness performed in Chapter 3. In Chapters 4 and 5, we focus on the design of a generalized ZMPC with mathematically proved convergence in the presence of bounded plant-model-mismatch. In Chapter 4 we proposed the controller with detailed mathematical proof presented. The performance of the proposed controller is validated on a continuous stirred tank with 2 system states, 1 system input, and one system output. Systematic guidelines on hyperparameter tuning is provided and the performance of various tuning parameter combination is shown via simulations. In Chapter 5, we modified the generalized robust ZMPC proposed in Chapter 4 such that it is applicable to a post-combustion carbon capture plant that has 103 system states, 2 system inputs, and 2 system outputs. The control performance of the proposed controller is shown to be more robust compared to a benchmark ZMPC.

6.2 Future Work

Physics-Informed Neural Network In this thesis, the investigation in NN-based model approximation approaches is purely data-driven. It is observed that if the number of inputs and outputs desired from the NN is high, the training complexity of the NN increases significantly with reduced prediction performance, making the approach impractical for detailed dynamic prediction of large-scale systems. However, if known first-principle knowledge can be intertwined into the NN model development, the structure of the NN can be optimized with less number of parameters to be identified in training. This could reduce the amount of data, the computational resources, and the effort required for training. Some existing results in this field include [84, 85, 86].

Transfer Learning for Online Update of Control Models Transfer learning is a powerful machine learning tool that employs a better-identified NN for a similar problem of interest with limited data available. It can significantly reduce the amount of data required and the computational effort for training, which makes it suitable for online model updates with limited time and measurement available. A neural network can be trained offline first with the historical data available. This NN can be reused and updated online with a time-triggering or task-triggering mechanism when a certain amount of time has passed or when the prediction error is larger than a preset threshold. Some existing results on this approach are [87, 88, 89]

Explicit Representation of Invariant Sets Approximating invariant sets for high dimensional systems is in general a very challenging task. A graph-based algorithm that obtains an inner-approximation of the invariant set was proposed in [90]. The algorithm discretizes the state space into small grids and determines the sectors that are inside the invariant set. Since this algorithm does not pre-specify the shape of the invariant set approximation, it is able to obtain a larger inner-approximation of the invariant set. However, the obtained set is not explicitly expressed and thus cannot be utilized directly in control applications

as constraints. A neural network could be a powerful tool in providing approximations of explicit mathematical expressions of these invariant sets and is worth investigating.

Bibliography

[1] E. S. Meadows, M. A. Henson, J. W. Eaton, and J. B. Rawlings, "Receding horizon control and discontinuous state feedback stabilization," *International Journal of Control*, vol. 62, no. 5, pp. 1217–1229, Nov. 1995. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/00207179508921593

[2] J. B. Rawlings, "Tutorial: model predictive control technology," in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*. San Diego, CA, USA: IEEE, 1999, pp. 662–676 vol.1. [Online]. Available: http://ieeexplore.ieee.org/document/782911/

[3] J. B. Rawlings, D. Angeli, and C. N. Bates, "Fundamentals of economic model predictive control," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. Maui, HI, USA: IEEE, Dec. 2012, pp. 3851–3861. [Online]. Available: http://ieeexplore.ieee.org/document/6425822/

[4] M. Ellis, J. Liu, and P. D. Christofides, *Economic Model Predictive Control*, ser. Advances in Industrial Control. Cham: Springer International Publishing, 2017. [Online]. Available: http://link.springer.com/10.1007/978-3-319-41108-8

[5] M. Diehl, R. Amrit, and J. B. Rawlings, "A Lyapunov Function for Economic Optimizing Model Predictive Control," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 703–707, Mar. 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5672577/

[6] M. Ellis, H. Durand, and P. D. Christofides, "A tutorial review of economic model predictive control methods," *Journal of Process Control*, vol. 24, no. 8, pp. 1156–1178, Aug. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0959152414000900

[7] P. O. M. Scokaert and J. B. Rawlings, "Feasibility issues in linear model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1649–1659, Aug. 1999. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/aic.690450805

[8] M. Askari, M. Moghavvemi, H. A. F. Almurib, and A. M. A. Haidar, "Stability of Soft-Constrained Finite Horizon Model Predictive Control," *IEEE Transactions on Industry Applications*, vol. 53, no. 6, pp. 5883–5892, Nov. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7959196/

[9] B. Grosman, E. Dassau, H. C. Zisser, L. Jovanovič, and F. J. Doyle, "Zone Model Predictive Control: A Strategy to Minimize Hyper- and Hypoglycemic Events," *Journal of Diabetes Science and Technology*, vol. 4, no. 4, pp. 961–975, Jul. 2010. [Online]. Available: http://journals.sagepub.com/doi/10.1177/193229681000400428

[10] A. H. González, P. S. Rivadeneira, A. Ferramosca, N. Magdelaine, and C. H. Moog, "Stable impulsive zone model predictive control for type 1 diabetic patients based on a long-term model," *Optimal Control Applications and Methods*, vol. 41, no. 6, pp. 2115–2136, Nov. 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/oca.2647

[11] S. Prívara, J. Široký, L. Ferkl, and J. Cigler, "Model predictive control of a building heating system: The first experience," *Energy and Buildings*, vol. 43, no. 2-3, pp. 564–572, Feb. 2011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378778810003749

[12] Y. Zhang, B. Decardi-Nelson, J. Liu, J. Shen, and J. Liu, "Zone economic model predictive control of a coal-fired boiler-turbine generating system," *Chemical Engineering Research and Design*, vol. 153, pp. 246–256, Jan. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0263876219304976

[13] Y. Mao, S. Liu, J. Nahar, J. Liu, and F. Ding, "Soil moisture regulation of agro-hydrological systems using zone model predictive control," *Computers and Electronics in Agriculture*, vol. 154, pp. 239–247, Nov. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0168169918304113

[14] Z. Huang, J. Liu, and B. Huang, "Model predictive control of agro-hydrological systems based on a two-layer neural network modeling framework," *International Journal of Adaptive Control and Signal Processing*, vol. 37, no. 6, pp. 1536–1558, 2023, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/acs.3586. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/acs.3586

[15] J. Zeng and J. Liu, "Economic Model Predictive Control of Wastewater Treatment Processes," *Industrial & Engineering Chemistry Research*, vol. 54, no. 21, pp. 5710–5721, Jun. 2015. [Online]. Available: https://pubs.acs.org/doi/10.1021/ie504995n

[16] J. Katz, B. Burnak, and E. N. Pistikopoulos, "The impact of model approximation in multiparametric model predictive control," *Chemical Engineering Research and Design*, vol. 139, pp. 211–223, Nov. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263876218304805

[17] V. B. Nguyen, S. B. Q. Tran, S. A. Khan, J. Rong, and J. Lou, "POD-DEIM model order reduction technique for model predictive control in continuous chemical processing," *Computers & Chemical Engineering*, vol. 133, p. 106638, Feb. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135419309895

[18] T. McKelvey, H. AkCay, and L. Ljung, "space-Based Multivariable System Identification from Frequency Response Data," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, vol. 41, no. 7, 1996.

[19] F. Felici, J.-W. Van Wingerden, and M. Verhaegen, "Subspace identification of MIMO LPV systems using a periodic scheduling sequence," *Automatica*, vol. 43, no. 10, pp. 1684–1697, Oct. 2007. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S000510980700194X

[20] A. Zhang and J. Liu, "Economic MPC of Wastewater Treatment Plants Based on Model Reduction," *Processes*, vol. 7, no. 10, p. 682, Oct. 2019. [Online]. Available: https://www.mdpi.com/2227-9717/7/10/682

[21] L. Zhang, D. Zhou, M. Zhong, and Y. Wang, "Improved closed-loop subspace identification based on principal component analysis and prior information," *Journal of Process Control*, vol. 80, pp. 235–246, Aug. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0959152418304360

[22] S. Saki and A. Fatehi, "Neural network identification in nonlinear model predictive control for frequent and infrequent operating points using nonlinearity measure," *ISA Transactions*, vol. 97, pp. 216–229, Feb. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0019057819303374

[23] Y. G. I. Acle, F. D. Freitas, N. Martins, and J. Rommes, "Parameter Preserving Model Order Reduction of Large Sparse Small-Signal Electromechanical Stability Power System Models," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2814–2824, Jul. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8640061/

[24] S. Ibrir, "A projection-based algorithm for model-order reduction with H 2 performance: A convex-optimization setting," *Automatica*, vol. 93, pp. 510–519, Jul. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109818301390

[25] J.-D. Diao, J. Guo, and C.-Y. Sun, "Event-triggered identification of FIR systems with binary-valued output observations," *Automatica*, vol. 98, pp. 95–102, Dec. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109818304497

[26] G. Liu, M.-l. Yu, L. Wang, Z.-y. Yin, J.-k. Liu, and Z.-r. Lu, "Rapid parameter identification of linear time-delay system from noisy frequency domain data," *Applied Mathematical Modelling*, vol. 83, pp. 736–753, Jul. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0307904X20301414

[27] B. D. O. Anderson, J. B. Moore, and R. M. Hawkes, "Model approximations via prediction error identification," *Automatica*, vol. 14, no. 6, pp. 615–622, 1978. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109878900511

[28] H. AKAIKE, "Maximum likelihood identification of Gaussian autoregressive moving average models," *Biometrika*, vol. 60, no. 2, pp. 255–265, Aug. 1973, _eprint: https://academic.oup.com/biomet/article-pdf/60/2/255/618377/60-2-255.pdf. [Online]. Available: https://doi.org/10.1093/biomet/60.2.255

[29] J. A. Vargas, W. Pedrycz, and E. M. Hemerly, "Improved learning algorithm for two-layer neural networks for identification of nonlinear systems," *Neurocomputing*, vol. 329, pp. 86–96, Feb. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231218311846

[30] J. Xu, Q. Tao, Z. Li, X. Xi, J. A. Suykens, and S. Wang, "Efficient hinging hyperplanes neural network and its application in nonlinear system identification," *Automatica*, vol. 116, p. 108906, Jun. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109820301047

[31] E. Koronaki, A. Nikas, and A. Boudouvis, "A data-driven reduced-order model of nonlinear processes based on diffusion maps and artificial neural networks,"

Chemical Engineering Journal, vol. 397, p. 125475, Oct. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1385894720314674

[32] Z. Wu, D. Rincon, and P. D. Christofides, "Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes," *Journal of Process Control*, vol. 89, pp. 74–84, May 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S095915241930825X

[33] Y. Zhou, S. Zheng, Z. Huang, L. Sui, and Y. Chen, "Explicit neural network model for predicting FRP-concrete interfacial bond strength based on a large database," *Composite Structures*, vol. 240, p. 111998, May 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S026382231932803X

[34] D. Angeli, R. Amrit, and J. B. Rawlings, "On Average Performance and Stability of Economic Model Predictive Control," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1615–1626, Jul. 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6099558/

[35] M. Heidarinejad, J. Liu, and P. D. Christofides, "Economic model predictive control of nonlinear process systems using Lyapunov techniques," *AIChE Journal*, vol. 58, no. 3, pp. 855–870, Mar. 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/aic.12672

[36] S. Liu and J. Liu, "Economic model predictive control with extended horizon," *Automatica*, vol. 73, pp. 180–192, Nov. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109816302540

[37] C. Liu, J. Gao, and D. Xu, "Lyapunov-based model predictive control for tracking of nonholonomic mobile robots under input constraints," *International Journal of Control, Automation and Systems*, vol. 15, no. 5, pp. 2313–2319, Oct. 2017. [Online]. Available: http://link.springer.com/10.1007/s12555-016-0350-x

[38] H. Makhamreh, M. Sleiman, O. Kukrer, and K. Al-Haddad, "Lyapunov-Based Model Predictive Control of a PUC7 Grid-Connected Multilevel Inverter," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 9, pp. 7012–7021, Sep. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8526525/

[39] S. Liu and J. Liu, "Economic Model Predictive Control with Zone Tracking," *Mathematics*, vol. 6, no. 5, p. 65, Apr. 2018. [Online]. Available: http://www.mdpi.com/2227-7390/6/5/65

[40] B. Decardi-Nelson and J. Liu, "Robust Economic Model Predictive Control with Zone Control," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 237–242, 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896321010211

[41] S. Liu, Y. Mao, and J. Liu, "Model-Predictive Control With Generalized Zone Tracking," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4698–4704, Nov. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8653956/

[42] X. Yin and J. Liu, "State estimation of wastewater treatment plants based on model approximation," *Computers & Chemical Engineering*, vol. 111, pp. 79–91, Mar. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0098135418300115

[43] S. K. Tiwary and R. A. Rutenbar, "On-the-Fly Fidelity Assessment for Trajectory-Based Circuit Macromodels," in *IEEE Custom Integrated Circuits Conference 2006*. San Jose, CA: IEEE, Sep. 2006, pp. 185–188. [Online]. Available: http://ieeexplore.ieee.org/document/4114935/

[44] M. Rewienski and J. White, "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 155–170, Feb. 2003. [Online]. Available: http://ieeexplore.ieee.org/document/1174092/

[45] B. Csáji, "Approximation with Artificial Neural Networks," Ph.D. dissertation, Eötvös Loránd University (ELTE), Jun. 2001.

[46] Z. Wu, A. Tran, D. Rincon, and P. D. Christofides, "Machine learning-based predictive control of nonlinear processes. Part I: Theory," *AIChE Journal*, vol. 65, no. 11, p. e16729, 2019, _eprint: https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.16729. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.16729

[47] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[48] P. D. Christofides, J. Liu, and D. M. d. l. Peña, *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications*, ser. Advances in Industrial Control Series. Springer-Verlag, London, England, 2011.

[49] R. J. Risbeck M.J., "Mpctools: Nonlinear model predictive control tools for casadi (python interface)," 2015. [Online]. Available: https://bitbucket.org/rawlings-group/mpc-tools-casadi

[50] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[51] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006. [Online]. Available: http://link.springer.com/10.1007/s10107-004-0559-y

[52] J. Alex, L. Benedetti, J. Copp, K. Gernaey, U. Jeppsson, I. Nopens, M.-N. Pons, L. Rieger, C. Rosen, and J.-P. Steyer, "Benchmark simulation model no. 1 (bsm1)," *Report by the IWA Taskgroup on Benchmarking of Control Strategies for WWTPs*, 01 2008.

[53] S. Mubako, S. Lahiri, and C. Lant, "Input–output analysis of virtual water transfers: Case study of California and Illinois," *Ecological Economics*, vol. 93, pp. 230–238, Sep. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921800913002012

[54] Y. Kim, R. G. Evans, and W. M. Iversen, "Evaluation of Closed-Loop Site-Specific Irrigation with Wireless Sensor Network," *Journal of Irrigation and Drainage Engineering*, vol. 135, no. 1, pp. 25–31, Feb. 2009. [Online]. Available: http://ascelibrary.org/doi/10.1061/%28ASCE%290733-9437%282009%29135%3A1%2825%29

[55] M. S. Goodchild, K. D. Kühn, M. D. Jenkins, K. J. Burek, and A. J. Dutton, "A Method for Precision Closed-loop Irrigation Using a Modified PID Control Algorithm," *Sensors and Transducers*, vol. 188, no. 5, pp. 61–68, May 2015.

[56] S. K. Saleem, D. K. Delgoda, S. K. Ooi, K. B. Dassanayake, L. Liu, M. N. Halgamuge, and H. Malano, "Model Predictive Control for Real-Time Irrigation Scheduling," *IFAC Proceedings Volumes*, vol. 46, no. 18, pp. 299–304, Aug. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1474667015350011

[57] C. Lozoya, C. Mendoza, L. Mejía, J. Quintana, G. Mendoza, M. Bustillos, O. Arras, and L. Solís, "Model Predictive Control for Closed-Loop Irrigation," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 4429–4434, 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1474667016422962

[58] C. Guo and F. You, "A Data-Driven Real-Time Irrigation Control Method Based on Model Predictive Control," in *2018 IEEE Conference on Decision and Control (CDC)*. Miami Beach, FL: IEEE, Dec. 2018, pp. 2599–2604. [Online]. Available: https://ieeexplore.ieee.org/document/8619549/

[59] J. Nahar, S. Liu, Y. Mao, J. Liu, and S. L. Shah, "Closed-Loop Scheduling and Control for Precision Irrigation," *Industrial & Engineering Chemistry Research*,

vol. 58, no. 26, pp. 11 485–11 497, Jul. 2019. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.iecr.8b06184

[60] S. R. Sahoo, B. T. Agyeman, S. Debnath, and J. Liu, "Knowledge-Based Optimal Irrigation Scheduling of Agro-Hydrological Systems," *Sustainability*, vol. 14, no. 3, p. 1304, Jan. 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/3/1304

[61] L. A. Richards, "CAPILLARY CONDUCTION OF LIQUIDS THROUGH POROUS MEDIUMS," *Physics*, vol. 1, no. 5, pp. 318–333, Nov. 1931. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.1745010

[62] M. T. van Genuchten, "A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils," *Soil Science Society of America Journal*, vol. 44, no. 5, pp. 892–898, Sep. 1980. [Online]. Available: http://doi.wiley.com/10.2136/sssaj1980.03615995004400050002x

[63] R. F. Carsel and R. S. Parrish, "Developing joint probability distributions of soil water retention characteristics," *Water Resources Research*, vol. 24, no. 5, pp. 755–769, May 1988. [Online]. Available: http://doi.wiley.com/10.1029/WR024i005p00755

[64] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[65] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," *arXiv:1506.00019 [cs]*, Oct. 2015, arXiv: 1506.00019. [Online]. Available: http://arxiv.org/abs/1506.00019

[66] Z. Huang, Q. Liu, J. Liu, and B. Huang, "A comparative study of model approximation methods applied to economic MPC," *The Canadian Journal of Chemical Engineering*, vol. 100, no. 8, pp. 1676–1702, 2022, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cjce.24398. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cjce.24398

[67] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org/

[68] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109899001132

[69] B. Decardi-Nelson and J. Liu, "Computing robust control invariant sets of constrained nonlinear systems: A graph algorithm approach," *Computers & Chemical Engineering*, vol. 145, p. 107177, Feb. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0098135420312205

[70] M. Höök and X. Tang, "Depletion of fossil fuels and anthropogenic climate change—A review," *Energy Policy*, vol. 52, pp. 797–809, Jan. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0301421512009275

[71] M. C. L. S. J. W. Chris Bataille, Henri Waisman and F. Jotzo, "The need for national deep decarbonization pathways for effective climate policy," *Climate Policy*, vol. 16, no. sup1, pp. S7–S26, 2016, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/14693062.2016.1173005. [Online]. Available: https://doi.org/10.1080/14693062.2016.1173005

[72] P. Babu, R. Kumar, and P. Linga, "Pre-combustion capture of carbon dioxide in a fixed bed reactor using the clathrate hydrate process," *Energy*, vol. 50, pp. 364–373, Feb. 2013.

[Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544212008262

[73] B. Decardi-Nelson, A. Akachuku, P. Osei, W. Srisang, F. Pouryousefi, and R. Idem, "A flexible and robust model for low temperature catalytic desorption of CO 2 from CO 2 -loaded amines over solid acid catalysts," *Chemical Engineering Science*, vol. 170, pp. 518–529, Oct. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0009250916307291

[74] A. Chansomwong, K. Zanganeh, A. Shafeen, P. Douglas, E. Croiset, and L. Ricardez-Sandoval, "Dynamic modelling of a CO2 capture and purification unit for an oxy-coal-fired power plant," *International Journal of Greenhouse Gas Control*, vol. 22, pp. 111–122, Mar. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1750583613004581

[75] J. Davison, L. Mancuso, and N. Ferrari, "Costs of CO2 Capture Technologies in Coal Fired Power and Hydrogen Plants," *Energy Procedia*, vol. 63, pp. 7598–7607, 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1876610214026095

[76] R. Sakwattanapong, A. Aroonwilas, and A. Veawab, "Behavior of Reboiler Heat Duty for CO2 Capture Plants Using Regenerable Single and Blended Alkanolamines," *Industrial & Engineering Chemistry Research*, vol. 44, no. 12, pp. 4465–4473, 2005, _eprint: https://doi.org/10.1021/ie050063w. [Online]. Available: https://doi.org/10.1021/ie050063w

[77] M. Panahi and S. Skogestad, "Economically efficient operation of CO2 capturing process. Part II. Design of control layer," *Chemical Engineering and Processing: Process Intensification*, vol. 52, pp. 112–124, Feb. 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0255270111002406

[78] T. Bankole, D. Jones, D. Bhattacharyya, R. Turton, and S. E. Zitney, "Optimal scheduling and its Lyapunov stability for advanced load-following energy plants with

CO2 capture," *Computers & Chemical Engineering*, vol. 109, pp. 30–47, Jan. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0098135417303800

[79] Z. He, M. H. Sahraei, and L. A. Ricardez-Sandoval, "Flexible operation and simultaneous scheduling and control of a CO2 capture plant using model predictive control," *International Journal of Greenhouse Gas Control*, vol. 48, pp. 300–311, May 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1750583615301122

[80] Z. Huang, J. Liu, and B. Huang, "Generalized robust MPC with zone-tracking," *Chemical Engineering Research and Design*, vol. 195, pp. 537–550, Jul. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0263876223003714

[81] B. Decardi-Nelson, S. Liu, and J. Liu, "Improving Flexibility and Energy Efficiency of Post-Combustion CO2 Capture Plants Using Economic Model Predictive Control," *Processes*, vol. 6, no. 9, 2018. [Online]. Available: https://www.mdpi.com/2227-9717/6/9/135

[82] B. Decardi-Nelson and J. Liu, "Robust Economic MPC of the Absorption Column in Post-Combustion Carbon Capture through Zone Tracking," *Energies*, vol. 15, no. 3, p. 1140, Feb. 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/3/1140

[83] B. Polyak and P. Shcherbakov, "Ellipsoidal approximations to attraction domains of linear systems with bounded control," in *2009 American Control Conference*, 2009, pp. 5363–5367.

[84] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: a review," *Acta Mechanica Sinica*, vol. 37, no. 12, pp. 1727–1738, Dec. 2021. [Online]. Available: https://link.springer.com/10.1007/s10409-021-01148-1

[85] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-Informed Neural Networks for Heat Transfer Problems," *Journal of Heat Transfer*, vol. 143, no. 6, p. 060801, Apr. 2021, _eprint: https://asmedigitalcollection.asme.org/heattransfer/article-pdf/143/6/060801/6688635/ht_143_06_060801.pdf. [Online]. Available: https://doi.org/10.1115/1.4050542

[86] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next," *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, Jul. 2022. [Online]. Available: https://doi.org/10.1007/s10915-022-01939-z

[87] P. Zhao, S. C. Hoi, J. Wang, and B. Li, "Online Transfer Learning," *Artificial Intelligence*, vol. 216, pp. 76–102, Nov. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0004370214000800

[88] B. Huang, T. Xu, Z. Shen, S. Jiang, B. Zhao, and Z. Bian, "Siamatl: Online update of siamese tracking network via attentional transfer learning," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 7527–7540, 2022.

[89] L. Liu, J. Wang, J. Li, and L. Wei, "An online transfer learning model for wind turbine power prediction based on spatial feature construction and system-wide update," *Applied Energy*, vol. 340, p. 121049, Jun. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306261923004130

[90] B. Decardi-Nelson and J. Liu, "An efficient implementation of graph-based invariant set algorithm for constrained nonlinear dynamical systems," *Computers and Chemical Engineering*, vol. 164, p. 107906, 2022.