Title
Machine Learning in Model Predictive Control, Operational Safety and Cybersecurity

Author
Wu, Zhe

Publication Date
2020

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Machine Learning in Model Predictive Control, Operational Safety and Cybersecurity

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Chemical Engineering

by

Zhe Wu

2020

ABSTRACT OF THE DISSERTATION

Machine Learning in Model Predictive Control, Operational Safety and Cybersecurity

by

Zhe Wu

Doctor of Philosophy in Chemical Engineering

University of California, Los Angeles, 2020

Professor Panagiotis D. Christofides, Chair

Big data is considered to play an important role in the fourth industrial revolution, which requires engineers and computers to fully utilize data to make smart decisions and improve the performance of industrial processes and of their control and safety systems. Traditionally, industrial process control systems rely on a (usually linear) data-driven model with parameters that are identified from industrial/simulation data, and in certain cases, for example, in profit-critical control loops, on first-principles models (with data-determined model parameters) that describe the underlying physico-chemical phenomena. However, modeling large-scale, complex nonlinear processes continues to be a major challenge in process systems engineering. Modeling is particularly important now and into the future, as process models are key elements of advanced model-based control systems, e.g., model predictive control (MPC) and economic MPC (EMPC).

Due to the wide variety of applications, machine learning models have great potential, yet, the development of rigorous and systematic methods for incorporating machine learning techniques in nonlinear process control and operational safety is in its infancy. Traditionally, operational safety of chemical processes has been addressed through process design considerations and through a hierarchical, independent design of control and safety systems. However, the consistent accidents throughout chemical process plant history (including several high profile disasters in

the last decade) have motivated researchers to design control systems that explicitly account for process operational safety considerations. In particular, a new design of control systems such as model predictive controllers (MPC) that incorporate safety considerations and can be coordinated with safety systems has the potential to significantly improve process operational safety and avoid unnecessary triggering of alarms systems, where machine learning techniques can be utilized to derive dynamic process models. However, the rigorous design of safety-based control systems poses new challenges that cannot be addressed with traditional process control methods, including, for example, proving simultaneous closed-loop stability and safety. On the other hand, cybersecurity has become increasingly important in chemical process industries in recent years as cyber-attacks that have grown in sophistication and frequency have become another leading cause of process safety incidents. While the traditional methods of handling cyber-attacks in control systems still rely partly on human analysis and mainly fall into the area of fault diagnosis, the intelligence of cyber-attacks and their accessibility to control system information has recently motivated researchers to develop cyber-attack detection and resilient operation control strategies to address directly cybersecurity concerns.

Motivated by the above considerations, this dissertation presents the use of machine learning techniques in model predictive control, operational safety and cybersecurity for chemical processes described by nonlinear dynamic models. The motivation and organization of this dissertation are first presented. Then, the use of machine learning techniques to develop data-driven nonlinear dynamic process models to be used in model predictive controllers is presented, followed by the discussion of real-time implementation with online learning of machine leaning models and of physics-based machine learning modeling methods. Subsequently, the MPC and economic MPC schemes that use control Lyapunov-barrier functions (CLBF) are presented in detail with rigorous analysis provided on their closed-loop stability, operational safety and recursive feasibility properties. Next, the development of machine-learning-based CLBF-MPC schemes is presented with process stability and safety analysis. Finally, the development of an integrated detection and control system for process cybersecurity is developed, in which several types of intelligent

cyber-attacks, machine learning detection methods and resilient control strategies are presented. Throughout the dissertation, the control methods are applied to numerical simulations of nonlinear chemical process examples to demonstrate their effectiveness and performance.

The dissertation of Zhe Wu is approved.

Philippe Sautet

Dante A. Simonetti

Paulo Tabuada

Panagiotis D. Christofides, Committee Chair

University of California, Los Angeles

2020

Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Professor Panagiotis D. Christofides, for his support and encouragement of both my technical work and also of my goals for my life over the years. Professor Christofides sets an example of excellence as a researcher, mentor, instructor, and role model. His mentoring on control theory, process/systems engineering, and life in general has made a profound impact on both my character and my career goals. I am also grateful for my family and friends and for their encouragement, support, and patience throughout my graduate career and throughout my life. In particular, I am grateful for my parents, Weilin Wu and Haiying Zhang, and my girlfriend, Yan Huang.

In addition, I would like to thank all of my colleagues with whom I have worked over the years in the Christofides research group, including Carlos Garcia, Yangyao Ding, Yichi Zhang, Yi ming Ren, Mohammed Alhajeri, Fahim Abdullah, Marquis Crose, Andres Aguirre, Anas Alanqar and Liangfeng Lao. I would particularly like to thank Professor Helen Durand, Professor Fahad Albalawi, Dr. Anh Tran, Dr. David Rincon, Dr. Zhihao Zhang, and Scarlett Chen with whom I have collaborated extensively and spent long hours working on papers together.

I would also like to thank Professor Philippe Sautet, Professor Dante A. Simonetti and Professor Paulo Tabuada for serving on my doctoral committee.

Financial support from UCLA Graduate Division Fellowship, UCLA Doctoral Dissertation Year Fellowship, the US Department of Energy (DOE) and the US National Science Foundation (NSF) is gratefully acknowledged, and my work could not have been done without this support.

Chapter 2 contains versions of: Z. Wu, A. Tran, D. Rincon and P. D. Christofides, "Machine Learning-Based Predictive Control of Nonlinear Processes. Part I: Theory", *AIChE J.*, **65**, e16729, 2019; Z. Wu, A. Tran, D. Rincon and P. D. Christofides, "Machine Learning-Based Predictive Control of Nonlinear Processes. Part II: Computational Implementation", *AIChE J.*, **65**, e16734, 2019;

Chapter 3 contains versions of: Z. Wu, and P. D. Christofides, "Economic Machine-Learning-Based Predictive Control of Nonlinear Systems", *Mathematics*, **7(6)**,

494, 20 pages, 2019.

Chapter 4 contains versions of: Z. Wu, D. Rincon and P. D. Christofides, "Real-Time Adaptive Machine-Learning-Based Predictive Control of Nonlinear Processes", *Ind. & Eng. Chem. Res.*, **59**, 2275-2290, 2020.

Chapter 5 contains versions of: Z. Wu, D. Rincon and P. D. Christofides, "Process Structure-based Recurrent Neural Network Modeling for Model Predictive Control of Nonlinear Processes," *J. Proc. Contr.*, **89**, 74-84, 2020.

Chapter 6 contains versions of: Z. Wu, F. Albalawi, Z. Zhang, J. Zhang, H. Durand and P. D. Christofides, "Control Lyapunov-Barrier Function-Based Model Predictive Control of Nonlinear Systems", *Automatica*, **109**, 108508, 2019; Z. Wu, and P. D. Christofides, "Handling Bounded and Unbounded Unsafe Sets in Control Lyapunov-Barrier Function-Based Model Predictive Control of Nonlinear Processes", *Chem. Eng. Res. & Des.*, **143**, 140-149, 2019; Z. Wu, H. Durand and P. D. Christofides, "Safe Economic Model Predictive Control of Nonlinear Systems", *Syst. & Contr. Lett.*, **118**, 69-76, 2018.

Chapter 7 contains versions of: Z. Wu, and P. D. Christofides, "Optimizing Process Economics and Operational Safety via Economic MPC Using Barrier Functions and Recurrent Neural Network Models", *Chem. Eng. Res. & Des.*, **152**, 455-465, 2019; Z. Wu, and P. D. Christofides, "Control Lyapunov-Barrier Function-Based Predictive Control of Nonlinear Processes Using Machine Learning Modeling", *Comp. & Chem. Eng.*, **134**, 106706, 2020; Z. Wu, D. Rincon and P. D. Christofides, "Real-time Machine Learning for Operational Safety of Nonlinear Processes via Barrier-Function Based Predictive Control", *Chem. Eng. Res. & Des.*, **155**, 88-97, 2020.

Chapter 8 contains versions of: Z. Wu, F . Albalawi, J. Zhang, Z. Zhang, H. Durand and P. D. Christofides, "Detecting and Handling Cyberattacks in Model Predictive Control of Chemical Processes," *Mathematics*, **6(10)**, 173, 22 pages, 2018; Z. Wu, S. Chen, D. Rincon and P. D. Christofides, "Post Cyber-Attack State Reconstruction for Nonlinear Processes Using Machine Learning," *Chem. Eng. Res. & Des.*, **159**, 248-261, 2020; S. Chen, Z. Wu and P. D. Christofides, "Cyber-attack Detection and Resilient Operation of Nonlinear Processes under Economic Model

Predictive Control," *Comp. & Chem. Eng.*, **136**, 106806, 2020.

Curriculum Vitae

Education

Zhejiang University **Sep. 2012 - July 2016**

B.S., Control Science & Engineering **Hangzhou, China**

Journal Publications

1. **Z. Wu**, D. Rincon and P. D. Christofides, "Process Structure-based Recurrent Neural Network Modeling for Model Predictive Control of Nonlinear Processes," *J. Proc. Contr.*, **89**, 74-84, 2020.

2. **Z. Wu**, D. Rincon and P. D. Christofides, "Real-Time Adaptive Machine-Learning-Based Predictive Control of Nonlinear Processes", *Ind. & Eng. Chem. Res.*, **59**, 2275-2290, 2020.

3. **Z. Wu**, and P. D. Christofides, "Control Lyapunov-Barrier Function-Based Predictive Control of Nonlinear Processes Using Machine Learning Modeling", *Comp. & Chem. Eng.*, **134**, 106706, 2020.

4. **Z. Wu**, D. Rincon and P. D. Christofides, "Real-time Machine Learning for Operational Safety of Nonlinear Processes via Barrier-Function Based Predictive Control", *Chem. Eng. Res. & Des.*, **155**, 88-97, 2020.

5. **Z. Wu**, F. Albalawi, Z. Zhang, J. Zhang, H. Durand and P. D. Christofides, "Control Lyapunov-Barrier Function-Based Model Predictive Control of Nonlinear Systems", *Automatica*, **109**, 108508, 2019.

6. **Z. Wu**, A. Tran, D. Rincon and P. D. Christofides, "Machine Learning-Based Predictive Control of Nonlinear Processes. Part I: Theory", *AIChE J.*, **65**, e16729, 2019.

7. **Z. Wu**, A. Tran, D. Rincon and P. D. Christofides, "Machine Learning-Based Predictive Control of Nonlinear Processes. Part II: Computational Implementation", *AIChE J.*, **65**, e16734, 2019.

8. **Z. Wu**, and P. D. Christofides, "Economic Machine-Learning-Based Predictive Control of Nonlinear Systems", *Mathematics*, **7(6)**, 494, 20 pages, 2019.

9. **Z. Wu**, and P. D. Christofides, "Handling Bounded and Unbounded Unsafe Sets in Control Lyapunov-Barrier Function-Based Model Predictive Control of Nonlinear Processes", *Chem. Eng. Res. & Des.*, **143**, 140-149, 2019.

13. **Z. Wu**, F. Albalawi, J. Zhang, Z. Zhang, H. Durand and P. D. Christofides, "Detecting and Handling Cyber-attacks in Model Predictive Control of Chemical Processes", *Mathematics*, **6(10)**, 173, 2019 2019.

10. **Z. Wu**, J. Zhang, Z. Zhang, F. Albalawi, H. Durand, M. Mahmood, P. Mhaskar and P. D. Christofides, "Economic model predictive control of stochastic nonlinear systems", *AIChE Journal*, **64(9)**, 3312-3322, 2018.

11. **Z. Wu**, H. Durand and P. D. Christofides, "Safe Economic Model Predictive Control of Nonlinear Systems", *Syst. & Contr. Lett.*, **118**, 69-76, 2018.

12. **Z. Wu**, A. Aguirre, A. Tran, H. Durand, D. Ni and P. D. Christofides, "Model Predictive Control of a Steam Methane Reforming Reactor Described by a CFD Model", *Ind. & Eng. Chem. Res.*, **56**, 6002-6011, 2017.

Chapter 1

Introduction

1.1 Motivation

The last decade has witnessed an explosive growth of data in modern industries, where a total volume of more than 1000 Exabytes data is estimated to be generated by machines and devices annually (e.g., [187]) Traditionally, industrial process control systems rely on a (usually linear) data-driven model with parameters that are identified from industrial/simulation data [38, 166], and in certain cases, for example, in profit-critical control loops, on first-principles models (with data-determined model parameters) that describe the underlying physico-chemical phenomena. However, modeling large-scale, complex nonlinear processes continues to be a major challenge in process systems engineering. Model quality depends on many factors, including, but not limited to model parameter estimation, model uncertainty, number of assumptions made in model development, dimensionality, model structure, and computational burden of solving the model in real-time (e.g., [47, 48]). In recent years, machine learning has attracted an increased level of attention in model identification. Among many machine learning techniques, recurrent neural networks (RNNs) have been widely-used for modeling general classes of nonlinear dynamical systems [34, 136]. In the last decade, along with the development of machine learning algorithms and computing resources/platforms, many open-source software libraries for machine learning

applications (e.g., [1, 33, 66]), such as Tensorflow and Keras, have been created, which have contributed to the broader use of machine learning techniques in classical engineering fields in addition to computer science and engineering (e.g., [9, 130, 155, 171, 180]). Specifically, feed-forward neural networks (FNN) and recurrent neural networks (RNN) (and their variants, see, for example, [34,53,107,136]) have demonstrated potential for use in model-based control systems since they are capable of modeling steady-state input-output nonlinear relationships and nonlinear dynamic behavior, respectively. To further improve the performance of machine learning models, on-line adaptation and training can be employed using real-time data sets collected from multiple sensors to reduce modeling error and account for model uncertainties, while implementation in parallel processing units could be employed to speed-up calculations for real-time tasks like process control and operational safety. Therefore, designing MPC systems that utilize machine learning modeling techniques to account in real-time for large data sets is a new frontier in control systems that will impact the next generation of industrial control systems.

Machine-learning-based MPC shows great potential in improving process operational safety, which is a long-standing research problem in optimal operation and control of dynamic systems and processes. The traditional approach to process operational safety is to employ a hierarchical approach as shown in Fig. 1.1. Specifically, a complete control and safety system used in industries includes basic process control systems (BPCS), alarm systems, emergency shut-down systems (ESS), and safety relief devices. Ideally, BPCS regulates process variables to their set-points while the layers of the safety system should not be activated regularly. When the BPCS fails to maintain the process variables within acceptable ranges due to, for example, equipment faults or unusually large process disturbances, alarms are triggered that alert operators so that actions can be taken to prevent further unsafe deviations. If the process variables subsequently further exceed allowable values, the ESS is triggered, which takes automatic and extreme actions such as forcing a valve to its fully open position to bring the process to a safer state of operation. Safety relief devices such as relief valves are used on vessels that can become highly pressurized quickly to prevent an explosion. Though safety systems and feedback control systems are critical to safe plant operation,

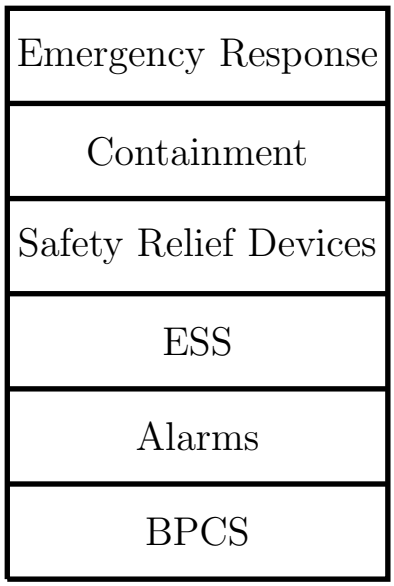


Figure 1.1: Control/safety system layers [91].

they act fully independently in the hierarchical multilevel system of Fig. 1.1 and are not integrated to yield cooperative actions to ensure both operational safety and economic performance. This has resulted in staggering profit losses for the chemical process industries. Therefore, it is necessary to coordinate the actions of process safety and control systems from both the ethical perspective of saving lives and property, and also from an economics standpoint for the chemical process industry. One potential solution is to incorporate safety considerations and safety system actions within optimization-based control schemes, e.g., model predictive control (MPC). While MPC has been widely-used in real-time operation of industrial chemical plants to optimize process performance accounting for closed-loop stability and control actuator constraints [46, 94, 99, 101, 125, 129], current MPC designs do not account for process safety considerations and actions and thus may lead to process operation in parts of the state space from which migration to an unsafe state may quickly occur. Therefore, a systematic methodology needs to be developed with rigorous analysis of process stability, operational safety and recursive feasibility to coordinate MPC systems and safety systems to ensure operational safety while achieving desired operation performance.

In addition to process operational safety, cybersecurity has become crucially important in recent years due to increasing risks of cyber-attacks with the development of modern communication

in industrial process controls and operations. Since both process safety and cybersecurity aim to prevent or mitigate events involving a loss of control of safety- and security-critical systems, the layers-of-protection analysis for safety systems can also be employed in the development of a defense-in-depth strategy for cyber-defense systems, where cybersecurity is incorporated into control network designs. Industrial control systems or supervisory control and data acquisition (SCADA) systems are usually large-scale, geographically dispersed and life-critical systems where embedded sensors, actuators and controllers are connected into a network to sense and control the physical devices [42]. The failure of cybersecurity can lead to unsafe process operation, and potentially to catastrophic consequences in chemical process industries, causing environmental damage, capital loss and human injuries. Cyber-attacks are essentially a series of computer actions to compromise the security of control systems (e.g., integrity, stability and safety) [41,44,118,186]. Among cyber-attacks, targeted attacks are severe threats for control systems because of their specific designs with the aim of modifying the control actions applied to a chemical process (for example, the Stuxnet worm aims to modify the data sent to a Programmable Logic Controller [27]). As the development of most of the existing detection methods still depends partly on human analysis, intelligent cyber-attacks that are process-aware and stealthy pose great challenges to the development of timely detection methods with high detection accuracy for modern industrial control system where cyber and physical components are closely interacted. Therefore, designing an advanced detection system and integrating it with MPC to handle cyber-attacks in safety-critical systems is a new frontier in control systems that will significantly improve security of chemical production.

1.2 Background

Machine learning has attracted an increased level of attention in model identification in recent years. Among many machine learning techniques, Recurrent neural networks (RNNs) have been widely-used for modeling general classes of nonlinear dynamical systems [34, 136]. Since

there exist feedback loops in RNN architectures that introduce the past information derived from earlier inputs into the current network, the information preserved in the internal states exhibits the memory of an RNN and leads to capturing dynamic behavior (from time-series of process state measurements) in a way conceptually similar to nonlinear dynamic models derived from first-principles. The history of recurrent neural networks can be traced back to the 1980s, when Hopfield networks were first created for pattern recognition [61]. Since then, many learning algorithms (e.g., supervised learning, unsupervised learning, and reinforcement learning) and modern RNN structures (e.g., Long short-term memory (LSTM), Gated recurrent unit (GRU) and Bidirectional recurrent neural networks) have been developed for various applications, for example, human visual pattern recognition. Machine learning techniques have now become increasingly popular in classical engineering fields in addition to computer science and engineering [9, 19, 124, 126, 130, 137, 171, 180]. Since neural networks are able to approximate any continuous function according to the universal approximation theorem [141], we can use recurrent neural networks to derive a nonlinear dynamic prediction model for MPC. MPC has been applied in real-time operation of industrial chemical plants to optimize process performance accounting for closed-loop stability, control actuator and safety constraints [46, 94, 99, 102, 125, 129]. Additionally, EMPC (in which the cost function does not have its minimum at a steady-state) may operate the system in a time-varying fashion (off steady-state) by optimizing an economic cost function accounting for stability constraints in one layer [14, 43, 104, 175]. One key requirement of MPC/EMPC is the availability of an accurate process model to predict states. Considering that in most cases it is difficult to obtain a first-principles model that captures complex, nonlinear behavior of a large-scale process, data-driven modeling [20, 148, 153, 157] has historically received significant attention in the context of MPC. Modeling through neural networks has proven to be successful in approximating nonlinear dynamical systems in [9, 80, 148]. Compared to polynomial approximation [4, 115, 152] which is generally easy to solve, neural networks may capture better 'difficult nonlinearities' via a richer class of nonlinear functions that can be learned effectively. Extensive research efforts have been made on RNN-based modeling, which also contributes to the

development of model-based control schemes that utilize data-driven models to predict process dynamics [116, 167, 184]. However, the existing works do not address real-time critical issues such as RNN-training and prediction error that ensures closed-loop stability under ML-MPC as well as computation time reduction for real-time implementation of machine-learning-based MPC.

Chemical process safety has traditionally been addressed through process design decisions (e.g., designing the process to be inherently safe in terms of its chemistry and physics [49, 58]) and control and safety system design decisions (e.g., adding sensors for critical process variables that trigger an alarm when a measurement outside of the desired range is obtained [91]). Inherently safer designs are achieved through four primary principles: minimize (reduce the quantity of hazardous substances used and stored by a process), substitute (utilize less hazardous process chemicals), moderate (dilute chemicals or change operating conditions), and simplify (choose designs with less complexity and less potential to create hazardous conditions when faults or errors occur) [52, 73]. However, it is not possible to eliminate all hazards at a plant, so a safety system, comprised of several independent layers, should be added (Fig. 1.1). While the hierarchical approach that utilizes control and safety systems independently for process safety has been successfully deployed in chemical process industries, the accidents throughout chemical plant history [76, 78, 90] have led some researchers to suggest that the philosophy used in the design of the control and safety system layers (i.e., designing barriers against specific unsafe scenarios using the safety system) is quite limited, particularly as economic considerations drive more optimized and integrated system designs [51, 55, 85, 110], and that a systems approach coordinating directly the actions of control and safety systems and analyzing closed-loop process operational safety should instead be used [7, 17, 37, 64, 82, 89, 154]. One step toward this systems approach is by incorporating safety considerations and safety system actions within the BPCS. However, the single-input/single-output controllers (e.g., proportional-integral-derivative controller (PID controller)) traditionally used within the BPCS cannot account for factors that are important to process safety such as multivariable interactions and state/input constraints. On the other hand, advanced model-based control methodologies such as model predictive control (MPC)

can account for these factors and thus can be integrated with safety considerations [82,94,99,125]. A large number of works in the MPC literature have addressed the robustness, performance and closed-loop stability of MPC (e.g., [26, 43, 57, 63, 94, 97, 101, 113, 188] and the references therein); but have not considered explicit safety considerations and safety system actions in their formulations.

Additionally, industrial process control systems rely heavily on information and communication technologies for automated operations. Particularly, industrial control systems integrate computation, networking and physical process components to seamlessly combine hardware and software resources for reliable operation and robust control. In more recent years, wireless networks and Internet communication are starting to replace or complement existing wired point-to-point communications in traditional large-scale process operations as well [35]. As these new developments bring efficiency to the existing system by enabling transmission of signals to remote locations without adding or altering the current hardwire infrastructure, heightened concern for security also arises [18]. A number of industrial cyber-attacks in recent years, such as the Stuxnet worm attacking Iran's nuclear centrifuges, the 2014 cyber-attack targeting a German steel mill, the 2015 cyber-attack on Ukraine's electric power grid, have all proven their detrimental physical impacts [75]. In recent years, cyber-security and cyber-defense have garnered increasing research interests with the rise of virtualization and big data [16, 39, 79], where machine learning techniques that can learn the system pattern from big data, provide a powerful tool to analyze industrial process data for the development of cyber-attack detection algorithms. In fact, machine learning has become increasingly popular in classical engineering fields in addition to computer science and engineering [9,19,124,126,130,137,155,166], and has shown promising potential for use in detection of cyber-attacks. For example, model-based fault diagnosis and classification in electric drives was carried out using a fault diagnostic neural network in [105] and automated fault detection and diagnosis of HVAC subsystems using hidden Markov models is shown in [163]. Using various machine learning classification methods, cyber-attacks on power systems are distinguished from process disturbances in [59], and a behavior-based intrusion detection

algorithm is developed to identify the types of attack [67]. Moreover, machine learning methods deployed for attack detection are presented in a number of literatures [24, 114, 135, 149, 164, 190]. While these recent literature contributions have demonstrated the feasibility of machine learning algorithms in anomaly management, the development of a protective safeguard through the integration of existing advanced control techniques (e.g., MPC) and online machine-learning-based detection algorithms to the multi-layer cyber-defense system that is of significant importance to next-generation smart manufacturing is still in its infancy.

1.3 Operational Safety and Cybersecurity of Chemical Processes

In this section, a chemical process example is presented to provide the motivation for developing novel control algorithms that account for operational safety and cybersecurity. In the first case study, the chemical process is operated in an off steady-state manner under economic model predictive control (EMPC) to optimize process economic performance. While the formal definition of EMPC will not be presented until the subsequent chapters, we can think of EMPC a predictive control scheme that optimizes operating strategy in real time to dynamically operate chemical processes in a bounded operating region in order to maximize process economic benefits accounting for time-varying economic factors, e.g., real-time energy and material pricing. However, in the case that the economically optimal regions include unsafe operating conditions, the time-varying operation of EMPC without accounting for safety region constraints may lead to unsafe operations when attempting to maximize process economic profits. The second case study considers the same chemical process and demonstrates the impact of cyber-attacks that compromise one of sensor measurements. Specifically, the system is normally operated at a pre-specified steady-state (either originally at the steady-state or forced to the steady-state from another operating condition) under feedback-based tracking model predictive control (MPC) with secure sensor measurements of process variables, e.g., temperature and species concentration;

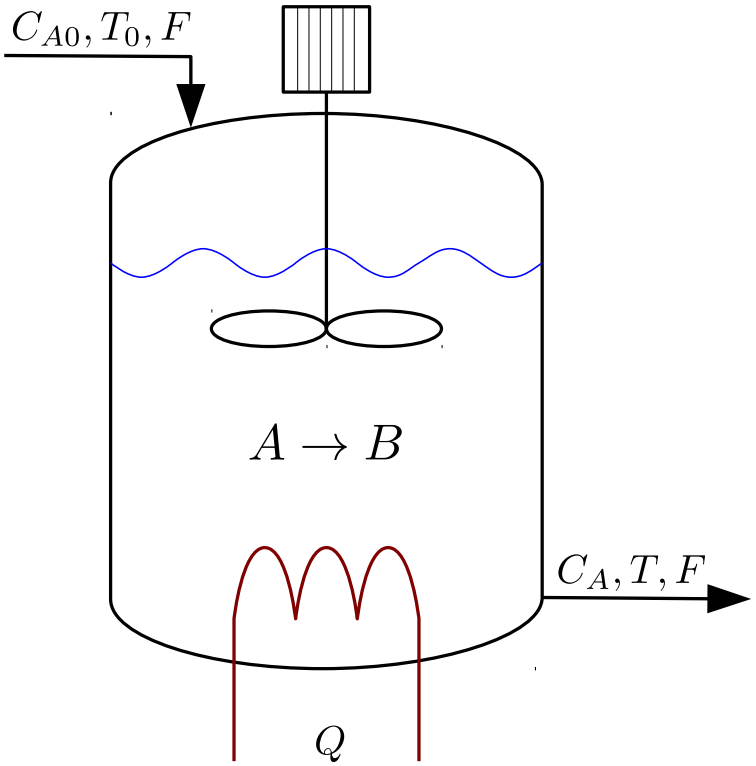


Figure 1.2: Diagram of a CSTR where a second-order reaction occurs that produces a desired product *B* from a reactant *A*.

however, it will be demonstrated that process stability is no longer guaranteed in the sense that the system may deviate from the steady-state and even leave the normal operating region when sensor measurements are tampered by cyber-attacks. The two case studies indicate the importance of having advanced control systems that account for process operational safety and cybersecurity, and have motivated much of the work contained in this dissertation. The chemical process example and the two case studies are provided below.

1.3.1 Continuously-Stirred Tank Reactor

Continuously stirred tank reactor (CSTR) with a second-order reaction is a well-established chemical engineering example that demonstrates performance improvement through time-varying operation. Specifically, we consider a non-isothermal CSTR where an elementary second-order reaction takes place that converts the reactant *A* to the desired product *B*. As shown in Fig. 1.2,

Table 1.1: Process parameters of the CSTR

$T_0 = 300$	K	$F = 5$	$\frac{m^3}{hr}$
$V = 1.0$	m^3	$E = 5 \times 10^4$	$\frac{kJ}{kmol}$
$k_0 = 8.46 \times 10^6$	$\frac{m^3}{kmol hr}$	$\Delta H = -1.15 \times 10^4$	$\frac{kJ}{kmol}$
$C_p = 0.231$	$\frac{kJ}{kgK}$	$R = 8.314$	$\frac{kJ}{kmolK}$
$\rho_L = 1000$	$\frac{kg}{m^3}$	$C_{A0s} = 4$	$\frac{kmol}{m^3}$
$Q_s = 0$	$\frac{kJ}{hr}$		

the reactant is fed to the reactor through a feedstock stream with concentration C_{A0}, volumetric flow rate F, and temperature T_0. The CSTR contents are assumed to be well-mixed, and the reactor is assumed to have a static liquid hold-up. The CSTR is equipped with a jacket to provide/remove heat to/from the reactor at a heat rate Q. Applying first principles and standard modeling assumptions, e.g., constant fluid density and heat capacity, which are denoted by ρ_L and C_p, respectively and Arrhenius rate dependence of the reaction rate on temperature, the following system of ordinary differential equations (ODEs) are developed to describe the evolution of the CSTR reactant concentration and temperature:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 \tag{1.1a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{1.1b}$$

where t is the time, C_A denotes the concentration of A in the reactor, T denotes the temperature of the reactor contents, V is the volume of the liquid hold-up in the reactor, k_0 is the rate constant, E is the reaction activation energy, ΔH is the enthalpy of reaction, and R is gas constant. The process parameter values are listed in Table 1.1.

The CSTR has an open-loop asymptotically stable steady-state at $[C_{As}\ T_s] = [1.22\ \frac{kmol}{m^3}\ 438\ K]$ and an unstable steady-state at $[C_{As}\ T_s]$=[1.95 $\frac{kmol}{m^3}$ 402K] which correspond to the steady-state input $[C_{A0s}\ Q_s] = [4\ \frac{kmol}{m^3}\ 0\ \frac{kJ}{hr}]$. In this example, the following production rate of B represents the

operating profit of the reactor:

$$r_B = k_0 e^{-\frac{E}{RT}} C_A^2 \tag{1.2}$$

The concentration C_{A0} of the reactant species A in the feed and the heat input/removal rate Q are used as the two manipulated inputs. Considering the physical bounds on C_{A0} and Q, the input constraints of the manipulated inputs are defined as follows: $|C_{A0} - C_{A0s}| \leq 3.5 \frac{kmol}{m^3}$ and $|Q - Q_s| \leq 5 \times 10^5 \frac{kJ}{hr}$.

1.3.2 Case Study: Process Operational Safety in EMPC

We first demonstrate the operational safety issue during the time-varying operation of the CSTR system of Eq. 1.1 under EMPC. The EMPC is designed to maximize process operating profits while maintaining the process states C_A and T within a bounded operating region around the stable steady-state at $(C_{As},\ T_s) = (1.22\ \frac{kmol}{m^3},\ 438\ K)$. Considering that thermal runaway may occur in CSTR systems when an increase in temperature changes the conditions in a way that causes a further increase in temperature, operating conditions of high temperature should be avoided in the dynamic operation. Additionally, to ensure that the operating profits of the CSTR system is maximized while the consumption of reactant A (i.e., inlet concentration C_{A0}) does not exceed its steady-state value, i.e., C_{A0s}, over the entire operating period, the following material constraint is employed in the optimization problem of EMPC:

$$\frac{1}{t_p} \int_0^{t_p} (C_{A0}(t) - C_{A0s}(t))\ dt = C_{A0s} \tag{1.3}$$

where t_p is the length of operation. It is trivial to show that without material constraint, the system will attempt to use the maximum amount of material for all times to maximize the production rate, which is not desired in an economical viewpoint as our goal is to determine the optimal strategy to distribute the material to the reactor.

The closed-loop simulation result under EMPC is shown in Fig. 1.3. It is demonstrated that the CSTR system is operated in an optimal time-varying manner where the process state, i.e.,

($C_A - C_{As}$, $T - T_s$), starts from the steady-state and first reaches the boundary of the operating region Ω_ρ which is a bounded set around the steady-state, then enters the unsafe region where temperature $T - T_s$ is greater than 47 K, and is finally driven back into the safe region due to the material constraint. Fig. 1.4b shows the accumulated operating profit profiles over the operating period $t_p = 1\ hr$, i.e., $\int_{t=0}^{t=t_p} r_B(\tau)d\tau$, under EMPC and under the steady-state operation (i.e., the CSTR is operated at the steady-state (C_{As} T_s) for all times), from which it is demonstrated that the time-varying operation of EMPC outperforms the steady-state operation in terms of economic performance. Additionally, the production rate profile for all the operating conditions within the operating region Ω_ρ is shown in Fig. 1.4a. It is observed that the optimal operating profit (i.e., the maximum value for the production rate of Eq. 1.2) is achieved near the right boundary of Ω_ρ, which explains why the state trajectory (blue dashed line) in Fig. 1.3 stays at the boundary of Ω_ρ for the majority of the operating time. From this case study, it is demonstrated that the EMPC scheme targeting process economic performance only is not able to achieve operational safety and economic optimality simultaneously, and therefore, a new EMPC design needs to be developed to incorporate safety considerations in its decision making to ensure operational safety while achieving desired economic performance.

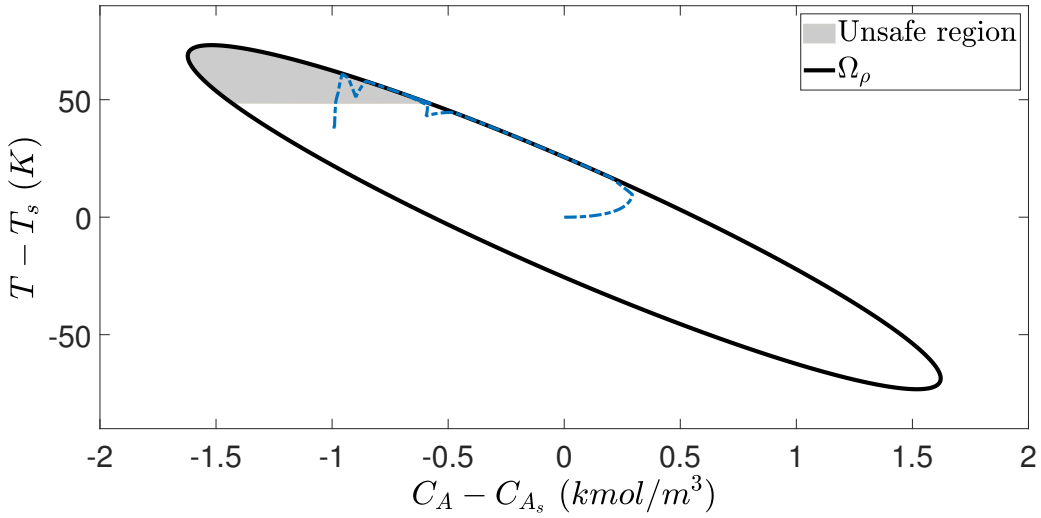
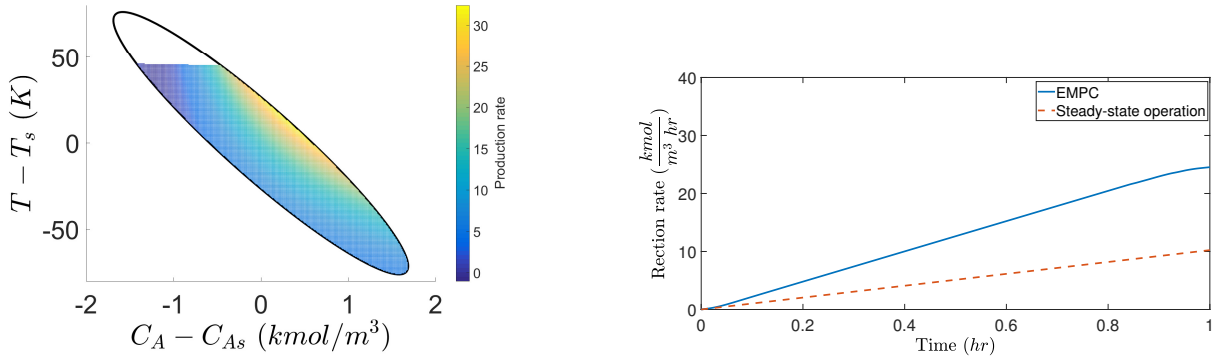


Figure 1.3: Closed-loop state trajectory for the CSTR under EMPC with the initial condition is at the steady-state, i.e., ($C_A(0) - C_{As}$, $T(0) - T_s$)=(0 $\frac{kmol}{m^3}$, 0K).



(a) (b)

Figure 1.4: (a) Production rate profile $k_0 e^{-E/RT} C_A^2 (\frac{kmol}{m^3\ hr})$ within the safe operating region of the CSTR, and (b) accumulated operating profits for the closed-loop CSTR under EMPC and steady-state operation, respectively.

1.3.3 Case Study: Cybersecurity in Tracking MPC

Consider the same CSTR system under a tracking MPC that aims to track the system state to an unstable steady-state $(C_{As}\ T_s)$=(1.95 $\frac{kmol}{m^3}$, 402K). The intrinsic unstable nature of the steady-state implies that without an appropriate controller, the CSTR system is not able to stably operate at the unstable steady-state; in other words, the steady-state inputs C_{A0s} and Q_s can neither stabilize the system at the steady-state if starting from another operating condition, nor maintain the system at the original steady-state under small perturbations. Therefore, a stabilizing controller such as proportional-integral-derivative controller or tracking MPC is required to operate the system at the unstable steady-state. We assume that the temperature sensor measurement for MPC is vulnerable to cyber-attacks in the sense that the measurement value that will be sent to the controller can be manipulated by attackers. Additionally, it is assumed that the intelligent cyber-attacks are process and controller behavior aware by having access to information on the CSTR operating region and existing alarms configured on the input and output ranges (in this particular example, alarms are triggered when the process state leaves the operating region). In this case, the controller that takes falsified temperature measurements will compute unreasonable control actions that may destabilize the system and lead to unsafe operations by driving process state off steady-state and ultimately out of the operating region. Fig. 1.5 shows the closed-loop simulation results for the

nominal CSTR system (i.e., under no attack) and the system under cyber-attacks. Specifically, the temperature sensor measurement is intruded by an intelligent cyber-attack that induces maximum disruption by setting the temperature value at its lower bound within the operating region since time $t = 0.03\ hr$. This type of cyber-attack is termed min-max cyber-attack, and will be formally defined in Chapter 8. As the temperature measurements (red trajectory) are maintained within the operating region Ω_ρ for all times as shown in Fig. 1.5, the min-max cyber-attack cannot be detected by conventional detection methods designed based on the boundary values. In Fig. 1.5, it is shown that starting from the initial condition ($C_A - C_{As}$, $T - T_s$)=(-1.2 $\frac{kmol}{m^3}$, $60K$), the closed-loop state trajectory (black, dashed line) is able to converge to the steady-state (C_{As}, T_s) under tracking MPC if no cyber-attack occurs. However, it is shown that without any detection system, the state trajectory (blue, solid line) with the same initial condition initially moves towards the origin following the same path under no attack. Then it starts deviating from the direction towards the origin and quickly leaves the operating region Ω_ρ due to incorrect control actions computed based on falsified temperature measurement (red, dash-dotted line) shown in Fig 1.6a. The system finally enters an unsafe region of extremely high temperature without being detected from sensor measurements, and therefore, alarm and emergency shut down systems based on other process variables are employed to prevent further unsafe deviations. Although the above cyber-attack only attacks one sensor, i.e., temperature sensor, it cannot be easily detected by control engineers by reading sensor measurements since the compromised values are bounded in the operating region at all times. Moreover, the cyber-attacks that are designed for industrial control systems will be more complicated in the sense that they can attack sensor networks in a coupled way, which makes it barely possible to detect from human analysis. Therefore, the example motivates the inquiry and theoretical developments of efficient data-based detection methods and resilient control strategies in the context of MPC systems that can eliminate the impact of cyber-attacks upon timely detection.

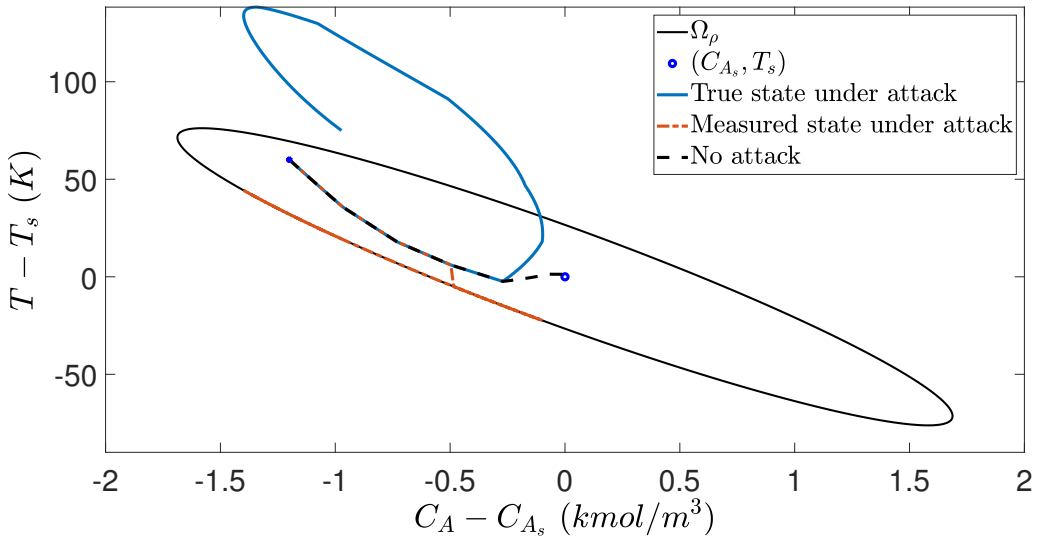
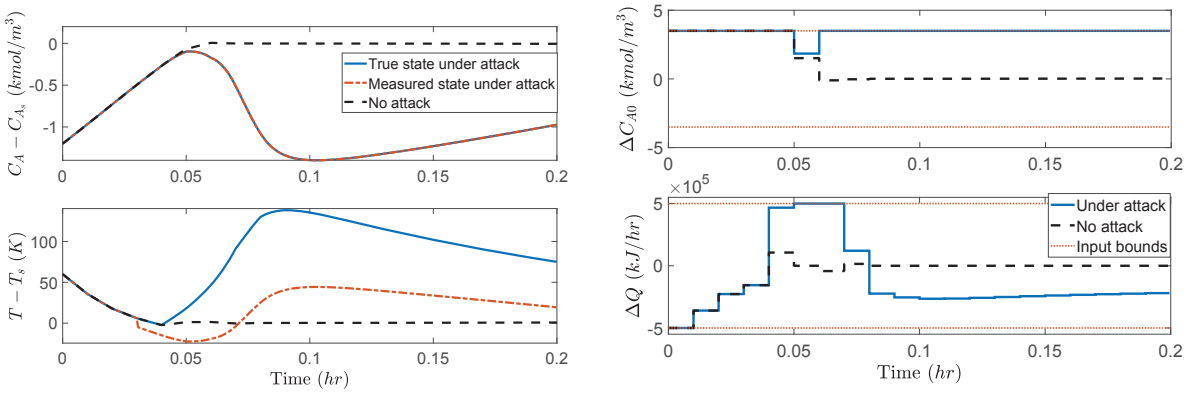


Figure 1.5: Closed-loop state trajectories for the CSTR under tracking MPC when the temperature sensor is under no attack, and under a min-max attack, respectively.



(a) (b)

Figure 1.6: (a) State and (b) input profiles for the CSTR under tracking MPC when the temperature sensor is under no attack, and under a min-max attack, respectively.

1.4 Dissertation Objectives and Structure

This dissertation presents control theoretic approaches to process operational safety and cybersecurity in the context of machine-learning-based model predictive control systems, and illustrates the applications of the proposed control methods using chemical process examples. Specifically, the objectives of this dissertation are summarized as follows:

1. To develop machine-learning-based model predictive control schemes by taking advantage of recurrent neural network modeling techniques with rigorous analysis provided on their closed-loop stability and safety properties.

2. To develop novel model predictive control systems using a new function termed control Lyapunov-barrier function to achieve guaranteed stability and safety properties and allow for recursive feasibility of MPCs.

3. To present a framework of integrating machine-learning-based detection systems with resilient control methods to handle cyber-attacks in industrial control systems.

4. To illustrate the applications of the developed control methods that account for process operational safety and cybersecurity to chemical process examples.

The dissertation is organized as follows. In Chapter 2 and Chapter 3, some definitions and preliminary results on stability and stabilization of nonlinear systems using Lyapunov's method are first provided. Then, the concept of recurrent neural networks (RNN) and a general framework to develop RNN models for nonlinear dynamic systems are introduced. The MPC and economic MPC schemes using RNN models for predicting system dynamics are developed, with sufficient conditions under which closed-loop stability and operational safety are derived. Computational implementation issues such as parallel computing and numerical approximation are also discussed. The methods are applied to the benchmark chemical reactor example.

In Chapter 4, real-time adaptive machine-learning-based predictive control scheme is developed by integrating online learning of RNN models. Event-triggered and error-triggered

mechanisms are designed to trigger online update within MPCs to capture process dynamics subject to time-varying disturbances using the most recent process operating data. The proposed methodology is applied to a chemical process example with disturbances under LMPC and LEMPC, respectively, to demonstrate the effectiveness of real-time implementation of machine-learning-based MPCs.

In Chapter 5, physics-based RNN modeling approaches are proposed for a general class of nonlinear dynamic process systems to improve prediction accuracy by incorporating a priori process knowledge. Three physics-based modeling approaches are presented, including a hybrid modeling method that integrates first-principles models and RNN models, a partially-connected RNN modeling method that designs the RNN structure based on a priori structural process knowledge, and a weight-constrained RNN modeling method that employs weight constraints in the optimization problem of the RNN training process. The proposed physics-based RNN models are utilized in model predictive controllers and applied to a chemical process network example to demonstrate their improved approximation performance compared to the fully-connected RNN model that is developed as a black box model.

In Chapter 6, the concept of operational safety in process control is introduced. Then, control Lyapunov-barrier function (CLBF) and CLBF-based MPC schemes are developed to integrate operational safety within control systems that optimize process performance. Based on that, the EMPC scheme that uses CLBF constraints to ensure stability and safety is developed to optimize process economic performance. Rigorous theoretical results of closed-loop stability, process operational safety and recursive feasibility of MPCs are developed. The methodologies are applied to chemical process examples with different types of unsafe operating regions to demonstrate their effectiveness.

In Chapter 7, machine-learning-based detection systems and resilient control schemes are developed to detect and mitigate the impact of stealthy cyber-attacks in MPC and EMPC systems. The construction method of data-based machine-learning detectors that can detect multiple classes of intelligent cyber-attacks is first presented. Several cyber-attack resilient control strategies are

subsequently developed to contain and eliminate the impact of cyber-attacks by reconfiguring the control system. The application to a benchmark multivariable nonlinear process example is presented to evaluate the ability of the integrated detection and mitigation scheme.

Chapter 8 summarizes the main results of the dissertation.

Chapter 2

Machine Learning in Model Predictive Control

2.1 Introduction

In industry linear empirical models are often used in controllers to operate the system at the steady-state. Despite the successful applications of linear empirical modeling in process industries, modeling nonlinear systems is always valuable to address systems-level task since chemical processes are inherently nonlinear, and thus, require nonlinear process models to improve the closed-loop performance of model-based controllers. Motivated by the above, machine learning, a method of data analysis that can be utilized to model nonlinear systems for model-based controllers, has attracted an increased level of attention in model identification in recent years. Specifically, among many machine learning techniques, recurrent neural networks (RNN) have been widely-used for modeling a general class of nonlinear dynamical systems. While feedforward neural networks use a one-way connectivity between units to model nonlinear systems, RNN architectures include feedback loops that introduce the past information derived from earlier inputs to the current network. Thus, the information preserved in the internal states exhibits the memory of an RNN and leads to capturing dynamic behavior in a way conceptually similar

to nonlinear state-space ordinary differential equation models. The history of recurrent neural networks can be traced back to the 1980s, when Hopfield networks were first created for pattern recognition [61]. With the rapid development of computational resources, machine learning techniques have become accessible in classical engineering fields in addition to computer science and engineering, and have shown many successful applications for modeling nonlinear systems, e.g., [30, 31, 80, 137, 148, 162, 177, 181, 182]. Moreover, given that a single data-driven model may not perfectly represent the process dynamics in the entire operating region, ensemble learning, a multi-model approach, has been utilized to combine the results of multiple models for complex systems. Specifically, ensemble learning uses several models that are obtained during a learning step to approximate particular outputs. Compared to a single model prediction, ensemble learning has demonstrated benefits in robustness and accuracy in solving classification and regression problems, e.g., [21, 96, 111, 120, 132, 147, 180, 189].

In this chapter and the next chapter, we present the use of machine learning techniques in developing data-driven models for MPCs that target process stability and economic optimality properties. The concept of recurrent neural networks and a general framework for developing RNN models for nonlinear dynamical systems are introduced. Subsequently, machine-learning-based MPC and EMPC schemes that use RNN models for prediction are presented with guaranteed closed-loop stability, followed by the discussion of ensemble learning of multiple RNN models in MPCs to improve prediction accuracy and the use of parallel computing to address the resulting computational implementation issues. The applications of machine learning-based control schemes to a chemical reactor demonstrate the ability of RNNs to model nonlinear dynamical systems and the effectiveness of the control schemes in stabilizing systems.

2.1.1 Notation

The set of real numbers is denoted by $\mathbf{R}$, and the set of nonnegative real numbers is denoted by $\mathbf{R}_+$. $\mathbf{R}^n$ is an n-dimensional real (Euclidean) space. The notation $|\cdot|$ is used to denote the Euclidean norm of a vector, and the notation $|\cdot|_Q$ denotes a weighted Euclidean norm of a vector

(i.e., $|x|_Q = \sqrt{x^T Q x}$ where Q is a positive definite matrix). The floor and ceiling functions, denoted as $\lfloor a \rfloor$ and $\lceil a \rceil$ for a scalar $a \in \mathbf{R}$, respectively, are the largest integer not greater than a and the smallest integer not less than a, respectively. x^T denotes the transpose of x. The variable t where $t \in \mathbf{R}_+$ is typically used to represent time, and thus, the notation $x(t) \in \mathbf{R}^n$ represents an n-dimensional time-dependent vector. An infinite sequence is denoted by $\{t_k\}_{k\geq 0}$, while a finite sequence is written as $\{t_i\}_{i=0}^N$ which describes the sequence: $t_0, t_1, \ldots, t_{N-1}, t_N$.

The notation $L_f V(x)$ denotes the standard Lie derivative of function $V(x)$ with respect to the vector field f, i.e., $L_f V(x) := \frac{\partial V(x)}{\partial x} f$. A scalar continuous function $V : \mathbf{R}^n \to \mathbf{R}$ is proper if the set $\{x \in \mathbf{R}^n \mid V(x) \leq k\}$ is compact for all $k \in \mathbf{R}$, or equivalently, V is radially unbounded in the sense that $\lim_{|x|\to+\infty} V(x) = +\infty$ holds. A function $V : \mathbf{R}^n \to \mathbf{R}_+$ is said to be positive definite with respect to $x \in \mathbf{R}^n$ if $V(x) > 0$ for all $x \in \mathbf{R}^n$ except that $V(x) = 0$ if and only if $x = 0$. A function, $V : \mathbf{R}^n \to (-\infty, 0]$, is negative definite (with respect to the origin) if $-V$ is positive definite. The set Ω_ρ is used to represent a level set of a scalar-valued positive definite function V: $\Omega_\rho := \{x \in \mathbf{R}^n \mid V(x) \leq \rho\}$ where $\rho > 0$.

For given positive real numbers β and ε, $\mathscr{B}_\beta(\varepsilon) := \{x \in \mathbf{R}^n \mid |x - \varepsilon| < \beta\}$ is an open ball around ε with radius of β. The relative complement of the set A in B is denoted by $A \backslash B := \{x \in A, x \notin B\}$. A function $f(\cdot)$ is of class $\mathscr{C}^1$ if it is continuously differentiable. A real-valued function $f(\cdot) : \mathbf{R}^n \to \mathbf{R}$ is called Lipschitz continuous if there exists a positive real constant k such that $|f(x) - f(y)| \leq k|x - y|$ holds for all $x, y \in \mathbf{R}^n$, and is called locally Lipschitz continuous if for each $y \in \mathbf{R}^n$, there exists an $L > 0$ such that f is Lipschitz continuous on the open ball $\mathscr{B}_L(y)$.

Given a set $\mathscr{D}$, the boundary, the closure, and the interior of $\mathscr{D}$ are denoted by $\partial\mathscr{D}$, $\overline{\mathscr{D}}$, and Int($\mathscr{D}$), respectively. A continuous function $\alpha : [0, a) \to \mathbf{R}_+$ is said to be of class $\mathscr{K}$ if it is strictly increasing and $\alpha(0) = 0$. The family of piecewise constant, right-continuous functions with period $\Delta > 0$ is denoted as $S(\Delta)$. If the vector-valued function $u(t) : [0, N\Delta) \to \mathbf{R}^m$ can be described by $u(t) = \bar{u}_i$, for $t \in [i\Delta, (i+1)\Delta)$, where N is a positive integer and $\bar{u}_i \in \mathbf{R}^m$, $i = 0, 1, ..., N-1$, then we say $u \in S(\Delta)$.

2.1.2 Class of Nonlinear Systems

The class of continuous-time nonlinear systems considered is described by the following system of first-order nonlinear ordinary differential equations:

$$\dot{x} = F(x,u,w) := f(x) + g(x)u + h(x)w,\ x(t_0) = x_0 \tag{2.1}$$

where $x \in D \subset \mathbf{R}^n$ is the state vector, $u \in U \subset \mathbf{R}^m$ is the manipulated input vector, and $w \in W$ is the disturbance vector, where $W := \{w \in \mathbf{R}^l \mid |w| \leq w_m,\ w_m \geq 0\}$. The control action constraint is defined by $u \in U := \{u_{\min} \leq u \leq u_{\max}\} \subset \mathbf{R}^m$, where u_{min} and u_{max} are the lower and upper bounds for the input vector, respectively. It is assumed that $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are sufficiently smooth vector and matrix functions of dimensions $n \times 1$, $n \times m$, and $n \times l$, respectively, with $f(0) = 0$. Therefore, the origin is a steady-state of the nominal system of Eq. 2.1 with $w(t) \equiv 0$. The measurement of $x(t)$ is assumed to be available for feedback at each sampling time $t_k = t_0 + k\Delta$, $k = 0, 1, \ldots$, where Δ is the sampling period.

2.1.3 Stabilization Via Control Lyapunov Function

Assumption 2.1. *We assume that there exists a stabilizing feedback controller $u = \Phi(x) \in U$ for the nominal system of Eq. 2.1 with $w(t) \equiv 0$ that renders the origin of the closed-loop system under continuous implementation of the controller exponentially stable in the sense that there exists a $\mathscr{C}^1$ Lyapunov function $V : D \to \mathbf{R}_+$ such that the following inequalities hold for all x in a neighborhood D around the origin:*

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2, \tag{2.2a}$$

$$\frac{\partial V(x)}{\partial x} F(x, \Phi(x), 0) \leq -c_3|x|^2, \tag{2.2b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq c_4|x| \tag{2.2c}$$

where c_i, $i = 1,2,3,4$ are positive real numbers.

The stability region Ω_ρ is designed as a level set of the Lyapunov function $V(x)$ within D, from which Eq. 2.2 is satisfied: $\Omega_\rho := \{x \in D \mid V(x) \leq \rho,\ \rho > 0\}$. It is readily shown that Ω_ρ is an invariant set since it holds that $\dot{V} \leq -c_3|x|^2$ under $u = \Phi(x) \in U$ for all $x \in \Omega_\rho$. The following control law is used to render the origin of the nominal system of Eq. 2.1 exponentially stable.

$$k_i(x) = \begin{cases} -\dfrac{p + \sqrt{p^2 + \gamma|q|^4}}{|q|^2} q_i & \text{if} \quad q \neq 0 \\ 0 & \text{if} \quad q = 0 \end{cases} \tag{2.3a}$$

$$\Phi_i(x) = \begin{cases} u_{min} & \text{if} \quad k_i(x) < u_{min} \\ k_i(x) & \text{if} \quad u_{min} \leq k_i(x) \leq u_{max} \\ u_{max} & \text{if} \quad k_i(x) > u_{max} \end{cases} \tag{2.3b}$$

where p denotes $L_f V(x)$, q_i denotes $L_{g_i} V(x)$, $q = [q_1 \cdots q_m]^T$, $f = [f_1 \cdots f_n]^T$, $g_i = [g_{i1} \cdots g_{in}]^T$, $(i = 1, 2, \cdots, m)$ and $\gamma > 0$. $k_i(x)$ of Eq. 2.3a represents the original Sontag control law without saturation. $\Phi_i(x)$ of Eq. 2.3b represents the i_{th} component of the saturated control law $\Phi(x)$ that accounts for the input constraint $u \in U$.

2.2 Recurrent Neural Network

We develop a recurrent neural network (RNN) model with the following form:

$$\dot{\hat{x}} = F_{nn}(\hat{x}, u) := A\hat{x} + \Theta^T y \tag{2.4}$$

where $\hat{x} \in D \subset \mathbf{R}^n$ is the RNN state vector and $u \in \mathbf{R}^m$ is the manipulated input vector. $y = [y_1, ..., y_n, y_{n+1}, ..., y_{m+n}] = [\sigma(\hat{x}_1), ..., \sigma(\hat{x}_n), u_1, ..., u_m] \in \mathbf{R}^{n+m}$ is a vector of both the network state $\hat{x}$ and the input u, where $\sigma(\cdot)$ is the nonlinear activation function (e.g., a sigmoid function $\sigma(x) = 1/(1 + e^{-x})$). A is a diagonal coefficient matrix, i.e., $A = diag\{-a_1, ..., -a_n\} \in \mathbf{R}^{n \times n}$, and $\Theta = [\theta_1, ..., \theta_n] \in \mathbf{R}^{(m+n) \times n}$ with $\theta_i = b_i[w_{i1}, ..., w_{i(m+n)}]$, $i = 1, ..., n$. a_i and b_i are constants. w_{ij} is the weight connecting the jth input to the ith neuron where $i = 1, ..., n$ and $j = 1, ..., (m + n)$.

Additionally, a_i is assumed to be positive such that each state $\hat{x}_i$ is bounded-input bounded-state stable. Throughout this chapter, we use x to represent the state of the nonlinear system of Eq. 2.1 and use $\hat{x}$ for the state of the RNN model Eq. 2.4. To simplify the discussion, the bias term is not included in the notation as it can be considered an additional constant input, and thus, does not affect the formulation of RNN of Eq. 2.4. Additionally, it is noted that the RNN model of Eq. 2.4 is an input-affine system, and therefore, it can be written in the form that is similar to Eq. 2.1:

$$\dot{\hat{x}} = \hat{f}(\hat{x}) + \hat{g}(\hat{x})u \tag{2.5}$$

where $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ can be derived from the coefficient matrices A and Θ in Eq. 2.4 and are assumed to be sufficiently smooth. Unlike the one-way connectivity between units in feedforward neural networks (FNN), RNNs have signals traveling in both directions by introducing loops in the network. As shown in Fig. 2.1, the states information derived from earlier inputs are fed back into the network, which exhibits a dynamic behavior. Consider the problem of approximating a class of continuous-time nonlinear systems of Eq. 2.1 by an RNN model of Eq. 2.4. Based on the universal approximation theorem for FNNs, it is shown in, e.g., [80, 141] that the RNN model with sufficient number of neurons is able to approximate any dynamic nonlinear system on compact subsets of the state-space for finite time. The following proposition demonstrates the approximation property of the RNN model:

Proposition 2.1 (Universal Approximation Theorem, c.f. [141])**.** *Consider the nonlinear system of Eq. 2.1 and the RNN model of Eq. 2.4 with the same initial condition* $x(0) = \hat{x}(0) = x_0 \in \Omega_\rho$. *For any* $\varepsilon > 0$ *and* $T > 0$*, there exists an optimal weight matrix* Θ^* *such that the state* $\hat{x}$ *of the RNN model of Eq. 2.4 with* $\Theta = \Theta^*$ *satisfies the following equation:*

$$\sup_{t\in[0,T]} |x(t) - \hat{x}(t)| \leq \varepsilon \tag{2.6}$$

Remark 2.1. *The RNN model of Eq. 2.4 is developed as a continuous-time network since it is utilized to approximate the input-output behavior of the continuous-time nonlinear system of*

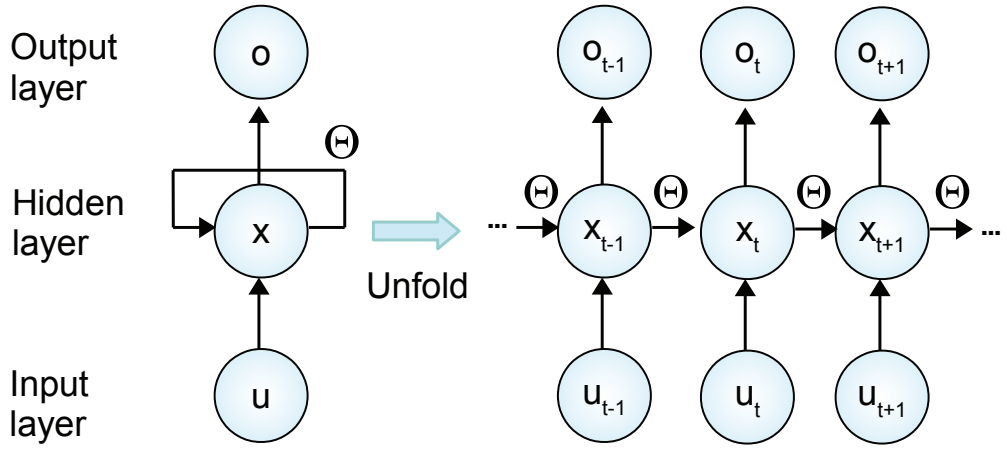


Figure 2.1: A recurrent neural network and its unfolded structure, where Θ is the weight matrix, x is the state vector, u is the input vector and o is the output vector (for the nonlinear system in the form of Eq. 2.1, the output vector is equal to the state vector).

Eq. 2.1. As discussed in [117], continuous-time RNNs have many advantages, for example, the well-defined derivative of the internal state with respect to time. However, it should be noted that the discrete-time RNNs can be equally well applied to model the nonlinear system of Eq. 2.1, where similar learning procedures and stability analysis can also be derived. Additionally, in this chapter, the RNN model of Eq. 2.4 and the controller that will be developed are both simulated in a sample-and-hold fashion with a sufficiently small sampling period Δ.

Remark 2.2. *To simplify the discussion, the RNN model of Eq. 2.4 is formulated as a one-hidden-layer RNN with n states to approximate the nonlinear system of n first-order ODEs of Eq. 2.1. However, the development of RNN models for approximation of the nonlinear system of Eq. 2.1 is not restricted to n-state, one-hidden-layer RNN model. Instead, to achieve a desired approximation performance of the nonlinear system of Eq. 2.1, a multi-layer RNN with a sufficient number of neurons is generally utilized. In that case, the RNN states $\hat{x} \in \mathbf{R}^n$ in Eq. 2.4 will be the last hidden layer or the output layer of an RNN.*

2.2.1 RNN Learning Algorithm

In this section, the RNN learning algorithm is developed to obtain the optimal weight matrix Θ^*, under which the error between the actual state $x(t)$ of the nominal system of Eq. 2.1 (i.e., $w(t) \equiv 0$) and the modeled states $\hat{x}(t)$ of the RNN of Eq. 2.4 is minimized. Although it is demonstrated in Proposition 2.1 that RNNs can approximate a broad class of nonlinear systems to any degree of accuracy, it is acknowledged that RNN modeling may not be perfect in many cases due to insufficient number of nodes or layers. Therefore, we assume that there exists a modeling error $\nu := F(x,u,0) - F_{nn}(\hat{x},u)$ between the nominal system of Eq. 2.1 and the RNN model of Eq. 2.4 with $\Theta = \Theta^*$. Since we focus on the system dynamics of Eq. 2.1 in a compact set Ω_ρ, from which the origin can be rendered exponentially stable using the controller $u = \Phi(x) \in U$, the RNN model is developed to capture the system dynamics for all $x \in \Omega_\rho$ and $u \in U$. It is noted that the modeling error $\nu(t)$ is bounded (i.e., $|\nu(t)| \leq \nu_m$, $\nu_m > 0$) since $x(t)$ and $u(t)$ are bounded. Additionally, to avoid the weight drift problem (i.e., the weights go to infinity during training), the weight vector θ_i is bounded by $|\theta_i| \leq \theta_m$, where $\theta_m > 0$. Following the methods in [80, 119], the RNN learning algorithm is developed to demonstrate that the state error $|e| = |\hat{x} - x|$ is bounded in the presence of the modeling error ν. Specifically, the RNN model is identified in the form of Eq. 2.4 and the nominal system of Eq. 2.1 (i.e., $w(t) \equiv 0$) can be expressed by the following equation:

$$\dot{x}_i = -a_i x_i + \theta_i^{*T} y + \nu_i,\ i = 1, ..., n \tag{2.7}$$

The optimal weight vector θ_i^* is defined as follows:

$$\theta_i^* := \arg \min_{|\theta_i| \leq \theta_m} \{ \sum_{k=1}^{N_d} |F_i(x_k, u_k, 0) + a_i x_k - \theta_i^T y_k| \} \tag{2.8}$$

where N_d is the number of data samples used for training. The state error is defined as $e = \hat{x} - x \in \mathbf{R}^n$. Based on Eq. 2.4 and Eq. 2.7, the time-derivative of state error is derived as follows:

$$\dot{e}_i = \dot{\hat{x}}_i - \dot{x}_i = -a_i e_i + \zeta_i^T y - \nu_i,\ i = 1, ..., n \tag{2.9}$$

where $\zeta_i = \theta_i - \theta_i^*$ is the error between the current weight vector θ_i and the unknown optimal weight vector θ_i^*. ν is the modeling error given by $\nu = F(x,u,0) - Ax - \Theta^* y$. The weight vector θ is updated during the training process as follows:

$$\dot{\theta}_i = -\eta_i y e_i,\ i = 1,...,n \tag{2.10}$$

where the learning rate η is a positive definite matrix. Based on the learning law of Eq. 2.10, the following theorem is established to demonstrate that the state error e remains bounded and is constrained by the modeling error ν.

Theorem 2.1 (c.f. [80, Theorem 4.1])**.** *Consider the RNN model of Eq. 2.4 of which the weights are updated according to Eq. 2.10. Then, the state error e_i and the weight error ζ_i are bounded, and there exist $\lambda \in \mathbf{R}$ and $\mu > 0$ such that the following inequality holds:*

$$\int_0^t |e(\tau)|^2 d\tau \leq \lambda + \mu \int_0^t |\nu(\tau)|^2 d\tau \tag{2.11}$$

Proof. We first define a Lyapunov function $\tilde{V} = \frac{1}{2}\sum_{i=1}^n (e_i^2 + \zeta_i^T \eta_i^{-1} \zeta_i)$. Based on Eq. 2.9, Eq. 2.10 and $\dot{\zeta}_i = \dot{\theta}_i$, the time-derivative of $\tilde{V}$ is derived as follows:

$$\begin{aligned} \dot{\tilde{V}} &= \sum_{i=1}^n (e_i \dot{e}_i + \eta_i^{-1} \zeta_i \dot{\zeta}_i) \\ &= \sum_{i=1}^n (-a_i e_i^2 - e_i \nu_i) \end{aligned} \tag{2.12}$$

It is noted that in the absence of modeling error (i.e., $\nu_i = 0$), it holds that $\dot{\tilde{V}} \leq 0$. Following the proof in [80], it is shown that the state error e_i and its time-derivative $\dot{e}_i$ are bounded for all times. Additionally, since $\tilde{V}$ is bounded from below and $\dot{\tilde{V}}$ is uniformly continuous implied by the fact that the second order derivative $\ddot{\tilde{V}}$ is bounded, it follows that $\dot{\tilde{V}} \to 0$ as $t \to \infty$ according to Barbalat's lemma* [106]. This implies that e_i ultimately converges to zero if there is no modeling error term

*Assume f is a function of time. Barbalat's lemma says: if $f(t)$ has a finite limit as $t \to \infty$, and if $\dot{f}$ is uniformly continuous, then $\dot{f}(t) \to 0$ as $t \to \infty$.

$-e_i\nu_i$ in Eq. 2.12. However, in the presence of modeling error $\nu_i \neq 0$, $\dot{\tilde{V}} \leq 0$ does not hold for all times. Therefore, based on Eq. 2.12, the following equation is further derived:

$$\begin{aligned}\dot{\tilde{V}} &= \sum_{i=1}^{n}(-\frac{a_i}{2}e_i^2 - \frac{1}{2}|\zeta_i|^2) + (\frac{1}{2}|\zeta_i|^2 - \frac{a_i}{2}e_i^2 - e_i\nu_i) \\ &\leq -\alpha\tilde{V} + \sum_{i=1}^{n}(\frac{1}{2}|\zeta_i|^2 - (\frac{a_i}{2}e_i^2 + e_i\nu_i + \frac{1}{2a_i}\nu_i^2) + \frac{1}{2a_i}\nu_i^2) \\ &\leq -\alpha\tilde{V} + \sum_{i=1}^{n}(\frac{1}{2}|\zeta_i|^2 + \frac{1}{2a_i}\nu_i^2)\end{aligned} \tag{2.13}$$

where $\alpha := \min\{a_i, 1/(\lambda_m),\ i = 1,...,n\}$ and λ_m represents the maximum eigenvalue of η_i^{-1}. Since the weight vector is bounded by $|\theta_i| \leq \theta_m$, it is derived that $\frac{1}{2}|\zeta_i|^2 \leq 2\theta_m^2$, and it follows that $\dot{\tilde{V}} \leq -\alpha\tilde{V} + \beta$, where $\beta := \sum_{i=1}^{n}(2\theta_m^2 + \nu_m^2/2a_i)$. Therefore, $\dot{\tilde{V}} \leq 0$ holds for all $\tilde{V} \geq V_0 = \beta/\alpha$, which implies that $\tilde{V}$ is bounded. From the definition of $\tilde{V}$, it is readily shown that e_i and ζ_i are bounded as well. Moreover, based on the fact that $\dot{\tilde{V}} \leq \sum_{i=1}^{n}(-\frac{a_i}{2}e_i^2 + \frac{1}{2a_i}\nu_i^2)$ derived from Eq. 2.13, we can also derive $\tilde{V}(t)$ as follows:

$$\begin{aligned}\tilde{V}(t) &\leq \tilde{V}(0) + \sum_{i=1}^{n}(-\frac{a_i}{2}\int_0^t e_i(\tau)^2 d\tau + \frac{1}{2a_i}\int_0^t \nu_i(\tau)^2 d\tau) \\ &\leq \tilde{V}(0) - \frac{a_{min}}{2}\int_0^t |e(\tau)|^2 d\tau + \frac{1}{2a_{min}}\int_0^t |\nu(\tau)|^2 d\tau\end{aligned} \tag{2.14}$$

where a_{min} is the minimum value of a_i, $i = 1,...,n$. Let $\lambda = \frac{2}{a_{min}}\sup_{t\geq 0}(\tilde{V}(0) - \tilde{V}(t))$ and $\mu = 1/a_{min}^2$. The relationship between $|e|$ and $|\mu|$ shown in Eq. 2.11 is derived as follows:

$$\begin{aligned}\int_0^t |e(\tau)|^2 d\tau &\leq \frac{2}{a_{min}}(\tilde{V}(0) - \tilde{V}(t)) + \frac{1}{a_{min}^2}\int_0^t |\nu(\tau)|^2 d\tau \\ &\leq \lambda + \mu\int_0^t |\nu(\tau)|^2 d\tau\end{aligned} \tag{2.15}$$

Therefore, it is guaranteed that the state error $|e|$ is bounded and is proportional to the modeling error $|\nu|$. Furthermore, it is noted that if there exists a positive real number $C > 0$ such that $\int_0^\infty |\nu(t)|^2 dt = C < \infty$, then it follows that $\int_0^\infty |e(t)|^2 dt \leq \lambda + \mu C < \infty$. Since $e(t)$ is uniformly continuous (i.e., $\dot{e}$ is bounded), it implies that $e(t)$ converges to zero asymptotically. □

Remark 2.3. *Since the weights may drift to infinity in the presence of modeling error, a robust learning algorithm named switching σ-modification approach was proposed in [80, 119] to adjust the weight such that the assumption that the weight vector θ_i is bounded by $|\theta_i| \leq \theta_m$ holds for all times. The switching σ-modification approach was then improved to be continuous in the considered compact set to overcome the problem of existence and uniqueness of solutions. The interested reader may refer to [80, 119] for further information.*

2.2.2 Development of RNN Model

In this section, we discuss how to develop an RNN model from scratch for a general class of nonlinear system of Eq. 2.1 within an operating region. Specifically, the development of a neural network model includes the generation of dataset and the training process.

2.2.2.1 Data generation

Open-loop simulations are first conducted to generate the dataset that captures the system dynamics for $x \in \Omega_\rho$ and $u \in U$ since we focus on the system dynamics of Eq. 2.1 in a compact set Ω_ρ with the constrained input vector $u \in U$. Given various $x_0 \in \Omega_\rho$, the open-loop simulations of the nominal system of Eq. 2.1 are carried out under various inputs u to obtain a large number of trajectories for finite time to sweep over all the values that (x,u) can take. However, it is noted that due to the limitation of computational resources, we may have to discretize the targeted region in state-space and the range of inputs with sufficiently small intervals in practice as shown in Fig. 2.2. Subsequently, time-series data from open-loop simulations can be separated into a large number of time-series samples with a shorter period P_{nn}, which represents the prediction period of RNNs. Lastly, the dataset is partitioned into training, validation and testing datasets. Additionally, it should be noted that we simulate the continuous system of Eq. 2.1 under a sequence of inputs $u \in U$ in a sample-and-hold fashion (i.e., the inputs are fed into the system of Eq. 2.1 as a piecewise constant function, $u(t) = u(t_k)$, $\forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$ and Δ is the sampling period). Then, the nominal system of Eq. 2.1 is integrated via explicit Euler method with a sufficiently

small integration time step $h_c < \Delta$.

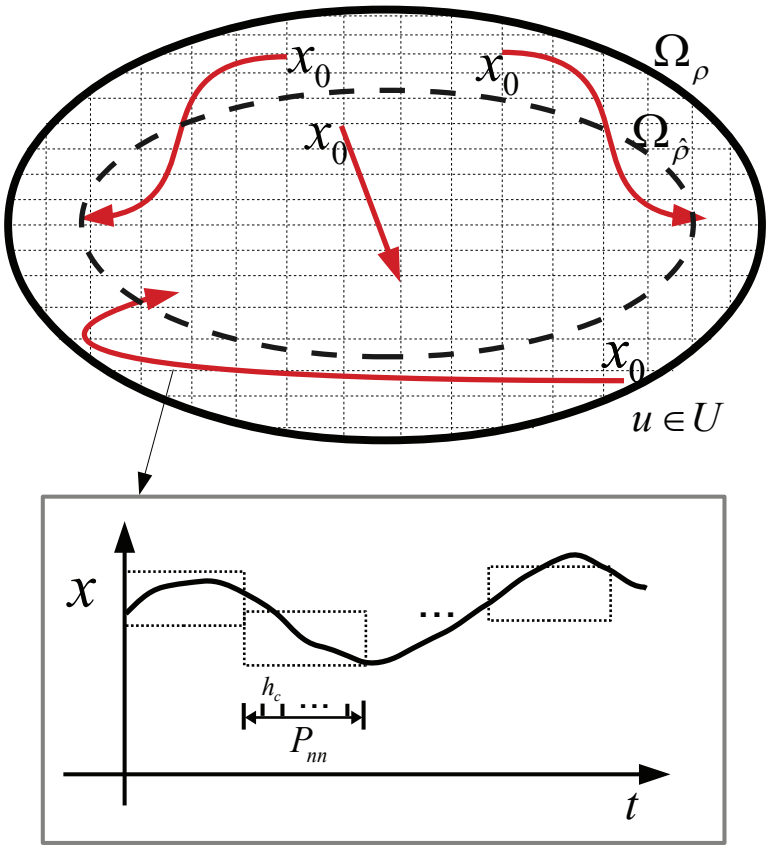


Figure 2.2: The schematic of discretization of the operating region Ω_ρ and the generation of training data for RNNs with a prediction period P_{nn} for all initial conditions $x_0 \in \Omega_\rho$, where h_c is the time interval between RNN internal states, Ω_ρ is the closed-loop stability region for the actual nonlinear system of Eq. 2.1 and $\Omega_{\hat{\rho}}$ is the closed-loop stability region characterized for the obtained RNN model.

2.2.2.2 Training process

Next, the RNN model is developed using a state-of-the-art application program interface (API), i.e., Keras [33]. The prediction period of RNN, P_{nn}, is chosen to be an integer multiple of the sampling period Δ such that the RNN model $F_{nn}(\hat{x}, u)$ can be utilized to predict future states for at least one sampling period by taking state measurement $x(t_k)$ and manipulated inputs $u(t_k)$ at time $t = t_k$. As shown in Fig. 2.2, the time interval between two consecutive internal states x_{t-1} and x_t within the prediction period P_{nn} for the unfolded RNN shown in Fig. 2.1 is chosen to be the integration

time step h_c used in open-loop simulations. Therefore, all the states between $t = 0$ and $t = P_{nn}$ with a step size of h_c are treated as the internal states and can be predicted by the RNN model. Based on the dataset generated from open-loop simulations, the RNN model of Eq. 2.4 is trained to calculate the optimal weight Θ^* of Eq. 2.8 by minimizing the modeling error between $F(x,u,0)$ and $F_{nn}(\hat{x},u)$. Furthermore, to guarantee that the RNN model of Eq. 2.4 achieves good performance in a neighborhood around the origin and has the same equilibrium point as the nonlinear system of Eq. 2.1, the modeling error is required to satisfy $|\nu| \leq \gamma|x| \leq \nu_m$ when the training process is completed.

Specifically, when we train an RNN using open-source neural-network libraries, for example, Keras, the optimization problem of minimizing the modeling error ν is solved using adaptive moment estimation method (i.e., Adam in Keras) with the loss function calculated by the mean squared error or the mean absolute percentage error between predicted states $\hat{x}$ and actual states x from training data. The optimal number of layers and neurons are determined through a grid search. Additionally, to avoid over-fitting of the RNN model, the training process is terminated once the modeling error falls below the desired threshold and the early-stopping condition is satisfied (i.e., the error on the validation set stops decreasing).

Remark 2.4. *In some cases training datasets may consist of noisy data or corrupt data, which could affect the training performance of RNNs in the following manners. On the one hand, noise makes it more challenging for RNNs to fit data points precisely. On the other hand, it is shown in literature, e.g., [21], that the addition of noise can also improve generalization performance and robustness of RNNs, and sometimes even lead to faster learning. Therefore, the neural network training with noise remains an important issue that needs further investigation. However, in this chapter, we perform open-loop simulations for the nominal system of Eq. 2.1 (i.e., $w(t) \equiv 0$), and thus, the RNN models are trained to approximate the dynamics of the nominal system of Eq. 2.1 within the closed-loop stability region $\Omega_{\hat{\rho}}$ based on the noise-free dataset.*

Remark 2.5. *The neural network modeling approach discussed in this chapter is a data-driven, black-box modeling approach that develops a nonlinear model of Eq. 2.4 to approximate the actual*

nonlinear system of Eq. 2.1 using massive amounts of process operating data. It is noted that neural network modeling is generally treated as a black-box modeling approach where no physical knowledge is utilized. However, in recent years, many researchers have also started to incorporate physical knowledge of systems into neural network formulations, trying to improve interpretability and optimality of neural network modeling. For example, it has been demonstrated in [87, 177] that the physics-based neural networks were able to improve the prediction performance when compared with a black-box neural network. For neural networks with incorporation of process knowledge, the interest reader is referred to [15, 70, 71, 86, 87, 138, 177]

2.2.3 Ensemble Regression Modeling

Since a single RNN may not perform perfectly over the entire operating region due to insufficient data and inappropriate ratio between the training dataset and validation dataset, the ensemble learning method, which is a machine learning process that combines multiple learning algorithms to obtain better prediction performance [96], is utilized to construct homogeneous ensemble regression models based on k-fold cross validation and the recurrent neural network (RNN) learning algorithm discussed in the previous section. Specifically, homogeneous ensemble regression models are derived from the ensemble learning method if a single base learning algorithm is used, while heterogeneous models are produced in the case of multiple learning algorithms. The reasons that ensemble regression models are able to improve the prediction performance are summarized in [96, 180] and are briefly stated as follows. First, a single RNN model that achieves a desired training accuracy may perform poorly in the region that lacks sufficient training data, while ensemble methods can reduce the risk of relying on a single flawed model by aggregating all candidate models. Second, the RNN learning algorithm is known to be a non-convex, NP-hard problem that can result in locally optimal solutions. Therefore, by using different starting initial weight matrices, ensemble learning methods might be able to avoid getting trapped in a local minimum and obtain a better set of weights for RNN to accurately predict output sequences. Third, in the case that the single regression model being trained cannot represent

the true target function, the ensemble learning methods might be able to provide some good approximation. Therefore, by introducing ensemble learning into the development of an empirical model for the nonlinear system of Eq. 2.1, the performance of ensemble regression models is expected to outperform that of a single RNN model in terms of reduced variability and improved generalization performance.

A rich collection of ensemble-based algorithms, e.g., Boosting, Bagging and Stacking, have been developed over the past few years [189]. In this chapter, the stacking method is used to combine the predictions of multiple regression models developed based on the RNN learning algorithm. It is noted that all the RNN models are developed to approximate the process dynamics for the entire operating region in this study. Specifically, following the approach of k-fold cross validation, the dataset generated from open-loop simulations is first split into k parts as shown in the dotted box in Fig. 2.3. Then, the RNN model with the general structure shown in Fig. 2.1 is trained using $k-1$ parts as training dataset and the remaining one as validation dataset to predict the nonlinear dynamics of the system of Eq. 2.1. Based on the training dataset, the detailed RNN structure is shown at the bottom of Fig. 2.3. In the input layer, the dimension of input nodes is $m+n$, where x_i, $i = 1,...,n$ represents the real-time state measurements and u_i, $i = 1,...,m$ represents the manipulated inputs at t_k. In the output layer, y_i, $i = 1,...,n$ report the estimation of the states after the prediction period P_{nn}. The output vector and the internal states can be used as the predicted states in the optimization problem of MPC that will be discussed in the next section. Additionally, two hidden layers are used in the RNN model with the optimal number of neurons determined through a grid search.

As shown in Fig. 2.3, the above training process is repeated to produce multiple RNN models using different $k-1$ sets as training dataset, and therefore, a total of k RNN models are developed based on k-fold cross validation. Subsequently, the final predicted states at $t = t_k + P_{nn}$ are calculated by a combiner algorithm that takes all the predictions generated by its constituent RNNs. In this study, we calculate the average of all prediction results from multiple RNNs as the final prediction results. Additionally, in Fig. 2.3, it is shown that normalizing and re-scaling

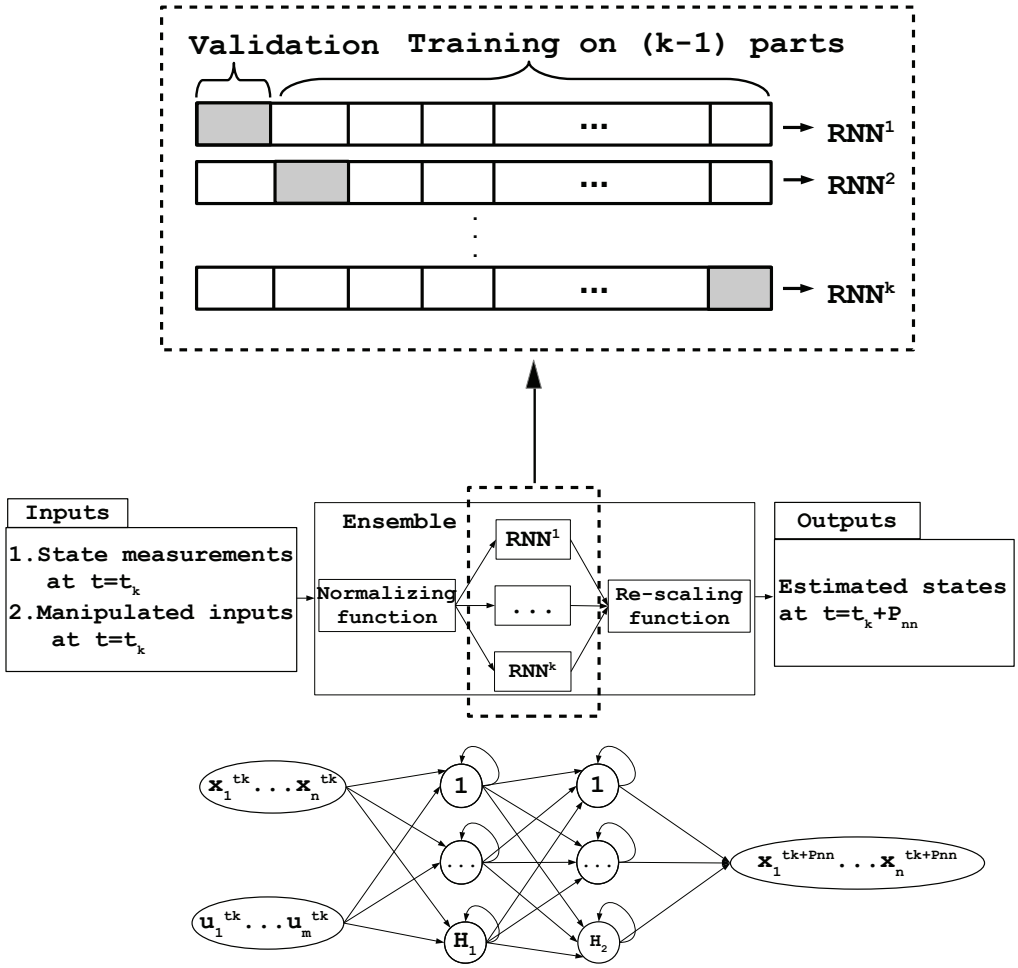


Figure 2.3: The structure of the ensemble regression models based on RNN learning algorithm and k-fold cross validation, where $x \in \mathbf{R}^n$ is the state vector, $u \in \mathbf{R}^m$ is the input vector, and H_1, H_2 are the number of neurons in hidden layers.

functions are employed before and after the ensemble of k RNN models. Specifically, the input vector, consisting of state measurements $x(t_k) \in \mathbf{R}^n$ and manipulated inputs $u(t_k) \in \mathbf{R}^m$ at $t = t_k$, is first normalized using the input statistics of the training dataset and fed into the cross-validated committee of RNNs. Subsequently, the output vector, which is the estimated states at $t = t_k + P_{nn}$, is generated by rescaling the average of the normalized predicted states using the output statistics of the training dataset.

Remark 2.6. *While ensemble regression models are utilized to improve prediction accuracy via the stacking method in this section, it is noted that averaging through the stacking method is not*

the only approach that can be applied here. For example, the bagging method that trains multiple models based on different subsets of the training dataset and calculates final predictive results through averaging or majority voting can be utilized to reduce the variance error [189]. The boosting method can also improve final predictive accuracy by adding more weights to incorrect prediction during the iterative training process. Additionally, further improvements may be achieved by combining results of multiple models that are derived using different machine learning methods through Bayesian model averaging, e.g., [22, 60].

2.3 Lyapunov-based MPC using Ensemble RNN Models

This section presents the formulation of Lyapunov-based MPC (LMPC) that incorporates the RNN model to predict future states with a stability analysis of the closed-loop system of Eq. 2.1. Specifically, the stability of the nonlinear system of Eq. 2.1 under a Lyapunov-based controller derived from the RNN model of Eq. 2.4 is first developed. Based on that, the RNN model of Eq. 2.4 is incorporated into the design of LMPC under sample-and-hold implementation of control actions to drive the closed-loop state to a small neighborhood around the origin.

2.3.1 Brief Overview of Model Predictive Control Methods

Before we present the formulation of RNN-based LMPC, a brief overview of model predictive control (MPC) is provided, followed by the design of Lyapunov-based MPC that ensures closed-loop stability only. The motivation for the use of MPC is due to the fact that the explicit feedback controller such as the Sontag control law of Eq. 2.3 may not be the optimal controller in general since process performance and system constraints are not explicitly taken into account. To overcome the shortcomings of explicit feedback controllers, MPC, also referred to as receding horizon control, has been proposed to control nonlinear processes and take process performance and constraints into considerations [26, 46, 94, 101, 125, 129]. MPC is essentially an on-line optimization-based control technique that optimizes a performance index or cost function over

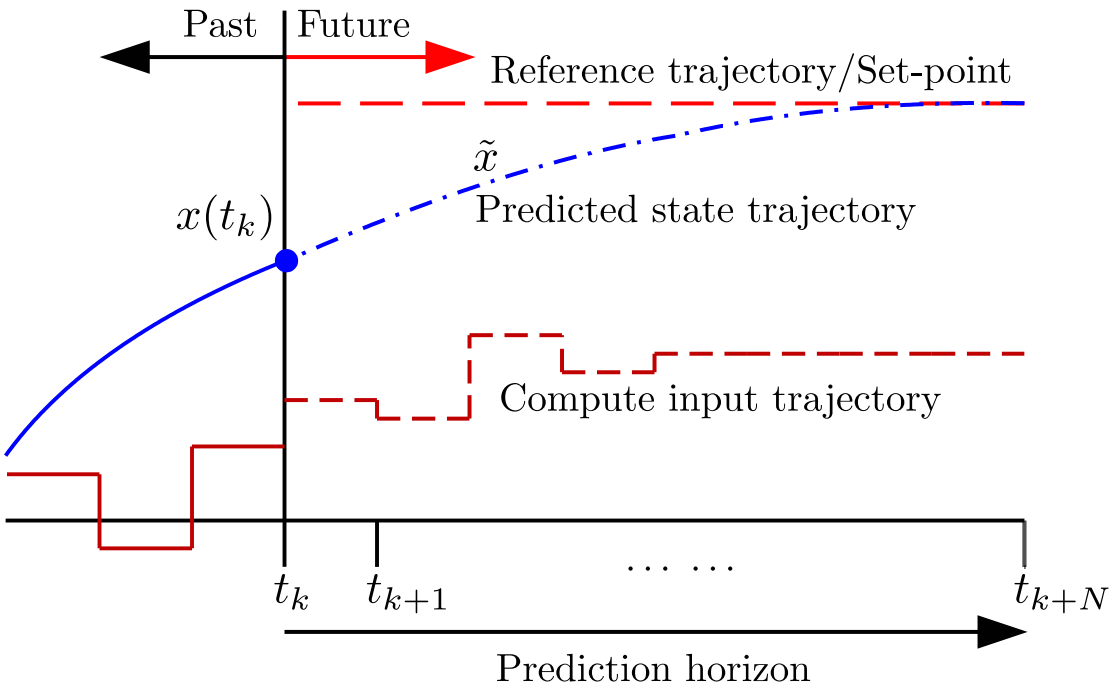


Figure 2.4: General concept for model predictive control (MPC).

a prediction horizon by taking advantage of a dynamic nominal process model, i.e., Eq. 2.1 with $w \equiv 0$, while accounting for system/process constraints. A brief overview of MPC is presented below.

2.3.1.1 Main components of MPC

As shown in Fig. 2.4, MPC typically optimizes the input trajectory (i.e., control actions) over the prediction horizon to track a set-point or a reference trajectory. The main components of MPC are listed as follows [43]:

1. A mathematical model of the process/system to predict the future evolution of the process/system over a time interval termed the prediction horizon.

2. A performance index or cost functional that maps the process/system (state, output and input) trajectories over the prediction horizon to a real number that is a measure of process performance. The cost functional is the objective function of the optimization problem.

3. Constraints on the process/system including restrictions on the control inputs, e.g., physical constraints on control actuators, and on system states/outputs, e.g., stability and safety constraints.

4. A receding horizon control approach to sampled-data implementation of controllers for continuous-time systems.

Compared to a proportional-integral-derivative (PID) controller, MPC has a number of advantages, which are summarized as follows. 1) As MPC allows the current timeslot (i.e., from t_k to t_{k+1} in Fig. 2.4) to be optimized, while taking future timeslots (i.e., the remaining part of the prediction horizon) into account, performance of closed-loop system such as energy consumption and speed of convergence to the set-point in chemical processes is improved under MPC compared to the PID controller that does not have predictive ability. 2) MPC has superior performance for processes with a large number of process variables (e.g., manipulated and controlled variables). 3) MPC allows constraints to be imposed on both manipulated and controlled variables while an integral windup often occurs in PID controllers as a limitation of physical systems. 4) Moreover, as the model accounts for inherent process characteristics (e.g., nonlinear behavior and multivariable interactions), MPC can accommodate a variety of process dynamics such as time delays, inverse response and inherent nonlinearities.

2.3.1.2 Process model

The mathematical model of the process/system is an essential element of an MPC controller as the prediction of process/system behavior is used in both MPC constraints and objective function. Traditionally, industrial MPCs utilized linear (empirical) process models, e.g., input-output model, state-space model, step, and impulse response models [129], because linear models can be considered good representations for many real processes over a small operating range, and are computationally cheap to solve. However, considering that chemical process are inherently nonlinear where nonlinearity comes from, for example, conservation of mass, momentum, and energy, nonlinear process models are preferred in MPCs to improve closed-loop control

performance when the processes are operated over a wide operating region. Additionally, MPC takes feedback information every sampling step to improve the poor performance due to linear models or imperfect nonlinear models used in the prediction.

2.3.1.3 Receding horizon implementation

MPC is implemented in a receding horizon manner in a way that the optimization problem is repeatedly solved on-line to compute the control actions. Specifically, real-time (continuous-time) is partitioned into discrete time steps called sampling times where the time between two consecutive sampling times, e.g., between t_k and t_{k+1} in Fig. 2.4, is called the sampling period. Process states/outputs are measured or estimated every sampling time to provide feedback information to MPC. At each sampling time, the MPC optimization problem is initialized with the state measurement or estimate, and is solved to compute the optimal control action(s) over the prediction horizon, from which, the first control action (i.e., for the first sampling period in the prediction horizon) will be applied to the closed-loop process/system. The horizon is moving one sampling period forward, and at the next sampling time, the MPC problem will be resolved with a new state measurement/estimate. The aforementioned steps are repeated until the end of operation. By adopting receding horizon implementation, the MPC is able to use process feedback information to improve closed-loop performance and compensate for process disturbances, modeling errors, and other forms of uncertainty. Additionally, it is noted that the solution to the infinite-horizon MPC (i.e., the MPC formulated with an infinite prediction horizon), assuming the solution exists, arguably gives the best solution as chemical processes are typically operated over long periods of time without a natural termination or shutdown time. However, to make the optimization problem of MPC more computationally tractable, the MPC is generally designed to be a finite-dimensional optimization problem with finite prediction horizon and optimized variables (i.e., control actions). Therefore, the receding horizon implementation also allows for a better approximation of the solution to the corresponding infinite-horizon MPC optimization problem.

2.3.1.4 Sample-and-hold implementation of controllers

Sample-and-hold implementation has been widely used in analog-to-digital converters that sample the voltage of a continuously varying analog signal and hold its value at a constant level for a period of time. As digital computers are commonly used in industrial control systems, sample-and-hold has also been utilized to integrate continuous-time physical systems with digital controllers. Specifically, given the continuous-time nonlinear system of Eq. 2.1, the following sampled time system is obtained:

$$x(t_{k+1}) \approx x(t_k) + \Delta f(x(t_k), u(t_k), w(t_k)) \tag{2.16}$$

where $t_k = k\Delta$, $k = 0,\ 1,\ ...$, and $\Delta > 0$ is the sampling period. u is a piecewise function of Δ, which means u holds constant within every sampling period, i.e., $u(t) = u(t_k),\ \forall t \in [t_k,\ t_{k+1})$. Additionally, the explicit Euler method is utilized to integrate the continuous-time system of Eq. 2.1 with a sufficiently small integration time step h_c ($0 < h_c \ll \Delta$) to provide a better approximation for the sampled time system of Eq. 2.16 by iteratively performing the following calculation within one sampling period:

$$x(t_k + h_c) \approx x(t_k) + h_c f(x(t_k), u(t_k), w(t_k)) \tag{2.17}$$

where Δ is an integer multiple of h_c. The state x, input u and disturbance w vectors in the function $f(x,u,w)$ are updated every h_c step, and thus, it takes $\frac{\Delta}{h_c}$ iterations of Eq. 2.17 to derive $x(t_{k+1}) := x(t_k + \Delta)$. Moreover, it should be noted that for the explicit Euler method, there exists an upper bound for the integration time step h_c to ensure numerical stability. Therefore, h_c is chosen to be a sufficiently small positive number in all the simulation studies throughout this dissertation.

2.3.1.5 MPC formulation

The MPC problem can be formulated as the following dynamic optimization problem:

$$\min_{u \in S(\Delta)} \quad \int_{t_k}^{t_{k+N}} l(\tilde{x}(\tau), u(\tau))\, d\tau \tag{2.18a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t), u(t), 0) \tag{2.18b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{2.18c}$$

$$(x(t), u(t)) \in Z,\ \forall\, t \in [t_k,\ t_{k+N}) \tag{2.18d}$$

where $\tilde{x}(t)$ is the predicted state trajectory, $S(\Delta)$ is the set of piecewise constant functions with period Δ, and N is the number of sampling periods in the prediction horizon. $l(x,u)$ is the cost function of MPC that represents process performance index; in tracking MPC that steers the system to the economically optimal steady-state or the economically optimal trajectory, the cost function is typically designed with a quadratic form that penalizes the deviations of the state and inputs from their corresponding optimal steady-state or reference values over the prediction horizon, i.e., $l(x,u) = |x|^2_{Q_1} + |u|^2_{Q_2}$, where Q_1, Q_2 are positive definite matrices that manage the trade-off between the speed of state convergence to the steady-state and the cost of control action. By designing the cost function in a quadratic form, the minimum value of the cost function is attained at the steady-state. The predicted state trajectory $\tilde{x}$ of Eq. 2.18b is obtained using the nominal dynamic model of Eq. 2.1 (i.e., $w \equiv 0$) under the piecewise constant input profile computed by the MPC. Eq. 2.18c defines the initial conditions for the nominal process system of Eq. 2.18b, which are obtained at each sampling period through a measurement. The constraints of Eq. 2.18d are the system/process constraints, where Z is assumed to be compact, which accounts, for example, state, input, and other process constraints. Throughout this dissertation, the term *MPC* will refer to tracking MPC that forces a system to steady-state, unless stated otherwise.

MPC is implemented in a receding horizon fashion to compute optimal control actions by solving the optimization problem of Eq. 2.18. Let $u^*(t)$ be the optimal solution of the optimization

problem of Eq. 2.18 over the prediction horizon $t \in [t_k, t_{k+N})$. We assume that the states of the closed-loop system are measured at each sampling time. Specifically, at the sampling time t_k, the problem of Eq. 2.18 is initialized with a state feedback measurement $x(t_k)$ and the problem is solved. After $u^*(t)$, where $t \in [t_k, t_{k+N})$, is obtained from the MPC optimization problem, only the first control action of $u^*(t)$, i.e., $u^*(t|t_k)$ defined for $t \in [t_k, t_{k+1})$, is sent to the control actuators to be applied over the next sampling period. Then, at the next instance of time $t_{k+1} := t_k + \Delta$, the problem is re-initialized with an updated state measurement and the optimization problem is solved again by rolling the horizon one sampling period forward.

However, since the MPC scheme of Eq. 2.18 is formulated with a finite prediction horizon, i.e., $N \neq \infty$, it should be noted that the MPC scheme of Eq. 2.18 may not be stabilizing, e.g., [94]. Therefore, to ensure stabilization of the closed-loop system with a finite N, additional constraints or variations to the cost function can be employed. For example, we can design an MPC with a sufficiently long prediction horizon, by incorporating terminal constraints, or using contractive constraints that will be discussed in more detail in the next section.

2.3.2 Lyapunov-based MPC

As previously mentioned, despite the well-characterized stability properties, the Lyapunov-based controllers are not guaranteed to be optimal as performance considerations are not accounted for in the calculation of control actions. Therefore, to improve process performance while ensuring stability of the closed-loop system, Lyapunov-based controller, i.e., a control law $\Phi(x)$ that satisfies asymptotic (exponential) stabilizability assumption, is utilized to design a contractive constraint in the formulation of MPC [36, 98, 99, 103]. The resulting tracking MPC is termed Lyapunov-based

MPC (LMPC) and is represented by the following optimization problem:

$$\min_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} l_t(\tilde{x}(\tau),u(\tau))\, d\tau \tag{2.19a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t),u(t),0) \tag{2.19b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{2.19c}$$

$$u(t) \in U,\ \forall\ t \in [t_k, t_{k+N}) \tag{2.19d}$$

$$\frac{\partial V(x(t_k))}{\partial x} f(x(t_k),u(t_k),0) \leq \frac{\partial V(x(t_k))}{\partial x} f(x(t_k),\Phi(x(t_k)),0),$$
$$\text{if } x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{min}} \tag{2.19e}$$

$$V(\tilde{x}(t)) \leq \rho_{min},\ \forall\ t \in [t_k, t_{k+N}),\ \text{if } x(t_k) \in \Omega_{\rho_{min}} \tag{2.19f}$$

where the notations follow those in Eq. 2.18. The objective function of Eq. 2.19 is the time-integral of $l_t(\tilde{x}(t),u(t))$ over the prediction horizon. The constraint of Eq. 2.19b is the nominal process model of Eq. 2.1 with $w(t) \equiv 0$ that is used to predict the states of the closed-loop system. Eq. 2.19c defines the input constraints applied over the entire prediction horizon. The constraint of Eq. 2.19e forces the closed-loop state to move towards the origin by decreasing the Lyapunov function value at least at the rate achieved by the Lyapunov-based controller $\Phi(x(t_k))$ at $t = t_k$, if $x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{min}}$. However, if $x(t_k)$ enters $\Omega_{\rho_{min}}$, which is a small neighborhood around the origin, the states predicted by the nonlinear model of Eq. 2.19b will be maintained in $\Omega_{\rho_{min}}$ for the entire prediction horizon under the constraint of Eq. 2.19f.

An illustration of the closed-loop state trajectory under LMPC is shown in Fig. 2.5, where x_s is the steady-state, Ω_ρ is the closed-loop stability region, Ω_{ρ_s} is a small level set close to the origin in which the Lyapunov function is not guaranteed to decay due to the sample-and-hold implementation of control actions and the effect of sufficiently small disturbances, and $\Omega_{\rho_{min}}$ is a small forward invariant set around the origin that ensures ultimate boundedness of the closed-loop state under LMPC. Closed-loop stability of the nonlinear system of Eq. 2.1 is guaranteed under the LMPC of Eq. 2.19 in the sense that for any initial condition $x_0 \in \Omega_\rho$, the closed-loop state is

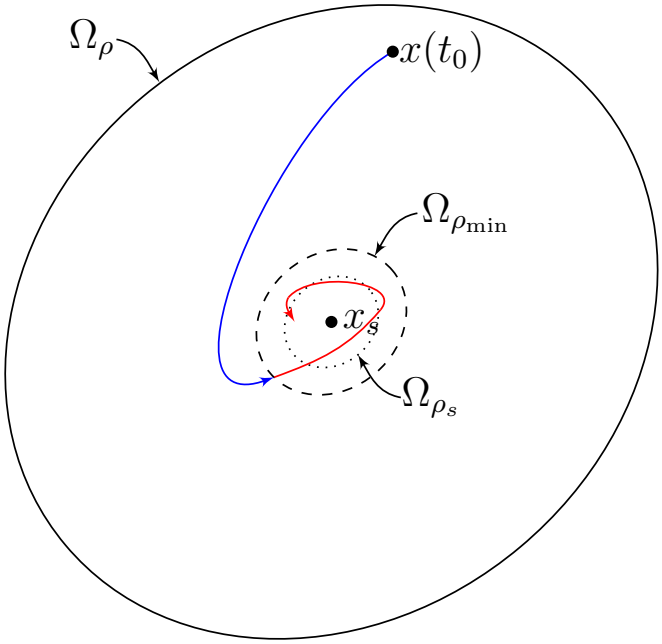


Figure 2.5: A state-space illustration of a closed-loop state trajectory under LMPC.

guaranteed to be bounded in Ω_ρ for all times and converge to a small neighborhood $\Omega_{\rho_{min}}$ of the origin and remains in it afterwards.

Since the LMPC of Eq. 2.19 needs a process model for predicting future states, in the following sections, we will incorporate RNN models in the formulation of LMPC and provide a rigorous a rigorous stability analysis for the closed-loop system under machine-learning-based MPC.

2.3.3 Lyapunov-based control using RNN models

A stabilizing feedback controller $u = \Phi_{nn}(x) \in U$ that can render the origin of the RNN model of Eq. 2.4 exponentially stable in an open neighborhood $\hat{D}$ around the origin is assumed to exist for the RNN model of Eq. 2.4 in the sense that there exists a $\mathscr{C}^1$ Control Lyapunov function $\hat{V}(x)$ such that the following inequalities hold for all x in $\hat{D}$:

$$\hat{c}_1|x|^2 \leq \hat{V}(x) \leq \hat{c}_2|x|^2, \tag{2.20a}$$

$$\frac{\partial \hat{V}(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x)) \leq -\hat{c}_3 |x|^2, \tag{2.20b}$$

$$\left| \frac{\partial \hat{V}(x)}{\partial x} \right| \leq \hat{c}_4 |x| \tag{2.20c}$$

where $\hat{c}_1$, $\hat{c}_2$, $\hat{c}_3$, $\hat{c}_4$ are positive constants, and $F_{nn}(x,u)$ represents the RNN model of Eq. 2.4. Similar to the characterization method of the closed-loop stability region Ω_ρ for the nonlinear system of Eq. 2.1, we first search the entire state-space to characterize a set of states $\hat{\phi}_u$ where the following inequality holds: $\dot{\hat{V}}(x) = \frac{\partial \hat{V}(x)}{\partial x} F_{nn}(x,u) < -k\hat{V}(x)$, $u = \Phi_{nn}(x) \in U$, $k > 0$. Starting from $\hat{\phi}_u$, the origin of the RNN model of Eq. 2.4 can be rendered exponentially stable under the controller $u = \Phi_{nn}(x) \in U$. The closed-loop stability region for the RNN model of Eq. 2.4 is defined as a level set of Lyapunov function inside $\hat{\phi}_u$: $\Omega_{\hat{\rho}} := \{x \in \hat{\phi}_u \mid \hat{V}(x) \leq \hat{\rho}\}$, where $\hat{\rho} > 0$. It is noted that the above assumption of Eq. 2.20 is the same as the assumption of Eq. 2.2 for the general class of nonlinear systems of Eq. 2.1 since the RNN model of Eq. 2.4 can be written in the form of Eq. 2.1 (i.e., $\dot{\hat{x}} = \hat{f}(\hat{x}) + \hat{g}(\hat{x})u$, where $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ are obtained from coefficient matrices A and Θ in Eq. 2.4). The assumptions of Eq. 2.2 and Eq. 2.20 are essentially the stabilizability requirements of the first-principles model of Eq. 2.1 and the RNN model of Eq. 2.4, respectively.

It is noted that $\Omega_{\hat{\rho}} \subseteq \Omega_\rho$ since the dataset for developing the RNN model of Eq. 2.4 is generated from open-loop simulations for $x \in \Omega_\rho$ and $u \in U$. Additionally, there exist positive constants M_{nn} and L_{nn} such that the following inequalities hold for all $x, x' \in \Omega_{\hat{\rho}}$ and $u \in U$:

$$|F_{nn}(x,u)| \leq M_{nn} \tag{2.21a}$$

$$\left| \frac{\partial \hat{V}(x)}{\partial x} F_{nn}(x,u) - \frac{\partial \hat{V}(x')}{\partial x} F_{nn}(x',u) \right| \leq L_{nn} |x - x'| \tag{2.21b}$$

Due to the model mismatch between the nominal system of Eq. 2.1 and the RNN model of Eq. 2.4, the following proposition is developed to demonstrate that the feedback controller $u = \Phi_{nn}(x) \in U$ is able to stabilize the nominal system of Eq. 2.1 if the modeling error is sufficiently small.

Proposition 2.2. *Under the assumption that the origin of the closed-loop RNN system of Eq. 2.4 is rendered exponentially stable under the controller $u = \Phi_{nn}(x) \in U$ for all $x \in \Omega_{\hat{\rho}}$, if there exists a*

positive real number $\gamma < \hat{c}_3/\hat{c}_4$ that constrains the modeling error $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma|x|$ for all $x \in \Omega_{\hat{\rho}}$ and $u \in U$, then the origin of the nominal closed-loop system of Eq. 2.1 under $u = \Phi_{nn}(x) \in U$ is also exponentially stable for all $x \in \Omega_{\hat{\rho}}$.

Proof. To demonstrate that the origin of the nominal system of Eq. 2.1 can be rendered exponentially stable $\forall x \in \Omega_{\hat{\rho}}$ under the controller for the RNN model of Eq. 2.4, we prove that $\dot{\hat{V}}$ for the nominal system of Eq. 2.1 can still be rendered negative under $u = \Phi_{nn}(x) \in U$. Based on Eq. 2.20b and Eq. 2.20c, the time-derivative of $\hat{V}$ is derived as follows:

$$\begin{aligned}
\dot{\hat{V}} =& \frac{\partial \hat{V}(x)}{\partial x} F(x, \Phi_{nn}(x), 0) \\
=& \frac{\partial \hat{V}(x)}{\partial x} (F_{nn}(x, \Phi_{nn}(x)) + F(x, \Phi_{nn}(x), 0) - F_{nn}(x, \Phi_{nn}(x))) \\
\leq& -\hat{c}_3|x|^2 + \hat{c}_4|x|(F(x, \Phi_{nn}(x), 0) - F_{nn}(x, \Phi_{nn}(x))) \\
\leq& -\hat{c}_3|x|^2 + \hat{c}_4\gamma|x|^2
\end{aligned} \tag{2.22}$$

If γ is chosen to satisfy $\gamma < \hat{c}_3/\hat{c}_4$, then it holds that $\dot{\hat{V}} \leq -\tilde{c}_3|x|^2 \leq 0$ where $\tilde{c}_3 = -\hat{c}_3 + \hat{c}_4\gamma > 0$. Therefore, the closed-loop state of the nominal system of Eq. 2.1 converges to the origin under $u = \Phi_{nn}(x) \in U$ for all $x_0 \in \Omega_{\hat{\rho}}$. □

Remark 2.7. *It should be noted that the RNN model of Eq. 2.4 that is trained on the dataset within the operating region Ω_ρ may not be stabilizable at the origin for all $x \in \Omega_\rho$ under $u = \Phi_{nn}(x) \in U$ due to model mismatch. Therefore, in this section, a new closed-loop stability region $\Omega_{\hat{\rho}}$ is characterized for the RNN model of Eq. 2.4 using the controller $\Phi_{nn}(x)$ in the form of Eq. 2.3. Subsequently, it is shown in Proposition 2.2 that the controller $\Phi_{nn}(x)$ also stabilizes the actual nonlinear system of Eq. 2.1 at the origin for all $x \in \Omega_{\hat{\rho}}$ provided that the modeling error is sufficiently small (i.e., $|\nu| \leq \gamma|x| < \hat{c}_3|x|/\hat{c}_4$). Since the closed-loop stability region $\Omega_{\hat{\rho}}$ for the RNN model of Eq. 2.4 guarantees the asymptotic stability of the origin for the nonlinear system of Eq. 2.1 under $u = \Phi_{nn}(x) \in U$, $\Omega_{\hat{\rho}}$ will be taken as the new operating region for the operation of model predictive control in the following sections.*

2.3.4 Sample-and-hold Implementation of Lyapunov-based Controller

Since the RNN model of Eq. 2.4 will be incorporated in Lyapunov-based MPC design, for which the control actions are implemented in sample-and-hold, in this section, we first derive the following propositions demonstrating the sample-and-hold properties of the Lyapunov-based controller $u = \Phi_{nn}(x)$ in the presence of bounded disturbances (i.e., $|w(t)| \leq w_m$). Specifically, the next proposition demonstrates that there exists an upper bound for the error between the state of the actual process of Eq. 2.1 in the presence of bounded disturbances (i.e., $|w(t)| \leq w_m$) and the state predicted by the RNN model of Eq. 2.4.

Proposition 2.3. *Consider the nonlinear system $\dot{x} = F(x,u,w)$ of Eq. 2.1 in the presence of bounded disturbances $|w(t)| \leq w_m$ and the RNN model $\dot{\hat{x}} = F_{nn}(\hat{x},u)$ of Eq. 2.4 with the same initial condition $x_0 = \hat{x}_0 \in \Omega_{\hat{\rho}}$. There exists a class $\mathscr{K}$ function $f_w(\cdot)$ and a positive constant κ such that the following inequalities hold $\forall x, \hat{x} \in \Omega_{\hat{\rho}}$ and $w(t) \in W$:*

$$|x(t) - \hat{x}(t)| \leq f_w(t) := \frac{L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1) \tag{2.23a}$$

$$\hat{V}(x) \leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{2.23b}$$

Proof. Let $e(t) = x(t) - \hat{x}(t)$ denote the error vector between the solutions of the system $\dot{x} = F(x,u,w)$ and the RNN model $\dot{\hat{x}} = F_{nn}(\hat{x},u)$. The time-derivative of $e(t)$ is obtained as follows:

$$\begin{aligned} |\dot{e}| &= |F(x,u,w) - F_{nn}(\hat{x},u)| \\ &\leq |F(x,u,w) - F(\hat{x},u,0)| + |F(\hat{x},u,0) - F_{nn}(\hat{x},u)| \end{aligned} \tag{2.24}$$

Following Eq. 2.21a, for all $\forall x, \hat{x} \in \Omega_{\hat{\rho}}$ and $w(t) \in W$, it is obtained that

$$\begin{aligned} |F(x,u,w) - F(\hat{x},u,0)| &\leq L_x|x(t) - \hat{x}(t)| + L_w|w(t)| \\ &\leq L_x|x(t) - \hat{x}(t)| + L_w w_m \end{aligned} \tag{2.25}$$

Additionally, the second term $|F(\hat{x},u,0) - F_{nn}(\hat{x},u)|$ in Eq. 2.24 represents the modeling error,

which is bounded by $|\nu| \leq \nu_m$ for all $\hat{x} \in \Omega_{\hat{\rho}}$. Therefore, based on Eq. 2.25 and the fact that $|F(\hat{x},u,0) - F_{nn}(\hat{x},u)| \leq \nu_m$, $\dot{e}(t)$ is bounded as follows:

$$\begin{aligned} |\dot{e}(t)| &\leq L_x|x(t) - \hat{x}(t)| + L_w|w_m| + \nu_m \\ &\leq L_x|e(t)| + L_w|w_m| + \nu_m \end{aligned} \tag{2.26}$$

Therefore, given the zero initial condition (i.e., $e(0) = 0$), the upper bound for the norm of the error vector is derived for all $x(t), \hat{x}(t) \in \Omega_{\hat{\rho}}$ and $|w(t)| \leq w_m$ as follows:

$$|e(t)| = |x(t) - \hat{x}(t)| \leq \frac{L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1) \tag{2.27}$$

Subsequently, $\forall x, \hat{x} \in \Omega_\rho$, Eq. 2.23b is derived based on the Taylor series expansion of $\hat{V}(x)$ around $\hat{x}$ as follows:

$$\hat{V}(x) \leq \hat{V}(\hat{x}) + \frac{\partial \hat{V}(\hat{x})}{\partial x}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{2.28}$$

where κ is a positive real number. Using Eq. 2.20a and Eq. 2.20c, it follows that

$$\hat{V}(x) \leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{2.29}$$

This completes the proof of Proposition 2.3. □

Subsequently, the following proposition is developed to demonstrate that the closed-loop state $x(t)$ of the actual process of Eq. 2.1 is bounded in $\Omega_{\hat{\rho}}$ for all times, and ultimately can be driven to a small neighborhood $\Omega_{\rho_{min}}$ around the origin under the controller $u = \Phi_{nn}(x) \in U$ implemented in a sample-and-hold fashion.

Proposition 2.4. *Consider the system of Eq. 2.1 under the controller $u = \Phi_{nn}(\hat{x}) \in U$ that is designed to stabilize the RNN system of Eq. 2.4 and meets the conditions of Eq. 2.20. The controller is implemented in a sample-and-hold fashion, i.e., $u(t) = \Phi_{nn}(\hat{x}(t_k))$, $\forall t \in [t_k, t_{k+1})$,*

where $t_{k+1} := t_k + \Delta$. *Let* $\varepsilon_s,\ \varepsilon_w > 0$, $\Delta > 0$ *and* $\hat{\rho} > \rho_{min} > \rho_{nn} > \rho_s$ *satisfy*

$$-\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{nn}M_{nn}\Delta \leq -\varepsilon_s \tag{2.30a}$$

$$-\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x^{'}M\Delta + L_w^{'}w_m \leq -\varepsilon_w \tag{2.30b}$$

and

$$\rho_{nn} := \max\{\hat{V}(\hat{x}(t+\Delta)) \mid \hat{x}(t) \in \Omega_{\rho_s}, u \in U\} \tag{2.31a}$$

$$\rho_{min} \geq \rho_{nn} + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}} f_w(\Delta) + \kappa(f_w(\Delta))^2 \tag{2.31b}$$

Then, for any $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, *the following inequality holds:*

$$\hat{V}(x(t)) \leq \hat{V}(x(t_k)),\ \forall t \in [t_k, t_{k+1}) \tag{2.32}$$

and the state $x(t)$ *of the nonlinear system of Eq. 2.1 is bounded in* $\Omega_{\hat{\rho}}$ *for all times and ultimately bounded in* $\Omega_{\rho_{min}}$.

Proof. Part 1 : Assuming $x(t_k) = \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, we first prove that the value of $\hat{V}(\hat{x})$ is decreasing under the controller $u(t) = \Phi_{nn}(x(t_k)) \in U$ for $t \in [t_k, t_{k+1})$, where $x(t)$ and $\hat{x}(t)$ denote the solutions of the nonlinear system of Eq. 2.1 in the presence of bounded disturbances and the RNN system of Eq. 2.4, respectively. The time-derivative of the $\hat{V}(\hat{x})$ along the trajectory $\hat{x}(t)$ of the RNN model of Eq. 2.4 in $t \in [t_k, t_{k+1})$ is obtained as follows:

$$\begin{aligned}
\dot{\hat{V}}(\hat{x}(t)) =& \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{nn}(\hat{x}(t), \Phi_{nn}(\hat{x}(t_k))) \\
=& \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{nn}(\hat{x}(t_k), \Phi_{nn}(\hat{x}(t_k))) + \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{nn}(\hat{x}(t), \Phi_{nn}(\hat{x}(t_k))) \\
& - \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{nn}(\hat{x}(t_k), \Phi_{nn}(\hat{x}(t_k)))
\end{aligned} \tag{2.33}$$

Using Eq. 2.20a and Eq. 2.20b, the following inequality is obtained:

$$\dot{\hat{V}}(\hat{x}(t)) \leq -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{nn}(\hat{x}(t), \Phi_{nn}(\hat{x}(t_k))) - \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{nn}(\hat{x}(t_k), \Phi_{nn}(\hat{x}(t_k))) \tag{2.34}$$

Based on the Lipschitz condition of Eq. 2.21 and the fact that $\hat{x} \in \Omega_{\hat{\rho}}$, $u \in U$, the upper bound of $\dot{\hat{V}}(\hat{x}(t))$ is derived $\forall t \in [t_k, t_{k+1})$:

$$\begin{aligned} \dot{\hat{V}}(\hat{x}(t)) &\leq -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{nn}|\hat{x}(t) - \hat{x}(t_k)| \\ &\leq -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{nn}M_{nn}\Delta \end{aligned} \tag{2.35}$$

Therefore, if Eq. 2.30a is satisfied, the following inequality holds $\forall \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k, t_{k+1})$:

$$\dot{\hat{V}}(\hat{x}(t)) \leq -\varepsilon_s \tag{2.36}$$

By integrating the above equation over $t \in [t_k, t_{k+1})$, it is obtained that $V(\hat{x}(t_{k+1})) \leq V(\hat{x}(t_k)) - \varepsilon_s\Delta$. So far we have proved that for all $\hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, the state of the closed-loop RNN system of Eq. 2.4 is bounded in the closed-loop stability region $\Omega_{\hat{\rho}}$ for all times and moves towards the origin under $u = \Phi_{nn}(\hat{x}) \in U$ implemented in a sample-and-hold fashion.

However, Eq. 2.36 may not hold when $x(t_k) = \hat{x}(t_k) \in \Omega_{\rho_s}$, which implies that the state may leave Ω_{ρ_s} within one sampling period. Therefore, according to Eq. 2.31a, $\Omega_{\rho_{nn}}$ is designed to ensure that the closed-loop state $\hat{x}(t)$ of the RNN model does not leave $\Omega_{\rho_{nn}}$ for all $t \in [t_k, t_{k+1})$, $u \in U$ and $\hat{x}(t_k) \in \Omega_{\rho_s}$ within one sampling period. If the state $\hat{x}(t_{k+1})$ leaves Ω_{ρ_s}, the controller $u = \Phi_{nn}(x(t_{k+1}))$ will drive the state towards Ω_{ρ_s} over the next sampling period since Eq. 2.36 is satisfied again at $t = t_{k+1}$. Therefore, the convergence of the state to $\Omega_{\rho_{nn}}$ for the closed-loop RNN system of Eq. 2.4 is proved for all $\hat{x}_0 \in \Omega_{\hat{\rho}}$. It remains to show that the closed-loop state of the actual nonlinear system of Eq. 2.1 can be bounded in $\Omega_{\hat{\rho}}$ for all times and ultimately bounded in a small neighborhood around the origin under the sample-and-hold implementation of the controller $u = \Phi_{nn}(x) \in U$.

Part 2 : Following the analysis performed for the RNN system of Eq. 2.4, we first assume $x(t_k) = \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and derive the time-derivative of $\hat{V}(x)$ for the nonlinear system of Eq. 2.1 (i.e., $\dot{x} = F(x,u,w)$) in the presence of bounded disturbances (i.e., $|w| \leq w_m$) as follows:

$$\begin{aligned}\dot{\hat{V}}(x(t)) =& \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\ =& \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) + \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\ & - \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0)\end{aligned} \tag{2.37}$$

From Eq. 2.22, $\frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \leq -\tilde{c}_3 |x(t_k)|^2$ holds for all $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$. Based on Eq. 2.20a and the Lipschitz condition in Eq. 2.21, the following inequality is obtained for $\dot{\hat{V}}(x(t))$ for all $t \in [t_k, t_{k+1})$ and $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$:

$$\begin{aligned}\dot{\hat{V}}(x(t)) \leq & -\frac{\tilde{c}_3}{\hat{c}_2} \rho_s + \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) - \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \\ \leq & -\frac{\tilde{c}_3}{\hat{c}_2} \rho_s + L_x^{'} |x(t) - x(t_k)| + L_w^{'} |w| \\ \leq & -\frac{\tilde{c}_3}{\hat{c}_2} \rho_s + L_x^{'} M \Delta + L_w^{'} w_m\end{aligned} \tag{2.38}$$

Therefore, if Eq. 2.30b is satisfied, the following inequality holds $\forall x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k, t_{k+1})$:

$$\dot{\hat{V}}(x(t)) \leq -\varepsilon_w \tag{2.39}$$

From Eq. 2.39, it is readily shown that Eq. 2.32 holds and the state of the closed-loop system of Eq. 2.1 is maintained in $\Omega_{\hat{\rho}}$ for all times. Also, it follows that the controller $u = \Phi_{nn}(x)$ is still able to drive the state of the actual nonlinear system of Eq. 2.1 towards the origin in every sampling period. Additionally, if $x(t_k) \in \Omega_{\rho_s}$, it is shown in *Part* 1 that the state of the RNN model of Eq. 2.4 is maintained in $\Omega_{\rho_{nn}}$ within one sampling period. Considering the bounded error between the state of the RNN of Eq. 2.4 model and the state of the nonlinear system of Eq. 2.1

given by Eq. 2.23a, there exists a compact set $\Omega_{\rho_{min}} \supset \Omega_{\rho_{nn}}$ that satisfies Eq. 2.31b such that the state of the actual nonlinear system of Eq. 2.1 does not leave $\Omega_{\rho_{min}}$ during one sampling period if the state of the RNN model of Eq. 2.4 is bounded in $\Omega_{\rho_{nn}}$. If the state $x(t)$ enters $\Omega_{\rho_{min}} \backslash \Omega_{\rho_s}$, we have shown that Eq. 2.39 holds, and thus, the state will be driven towards the origin again under $u = \Phi_{nn}(x)$ during the next sampling period. This completes the proof of Proposition 2.4 by showing that for any $x_0 = \hat{x}_0 \in \Omega_{\hat{\rho}}$, the closed-loop state trajectories of the nonlinear system of Eq. 2.1 are maintained in $\Omega_{\hat{\rho}}$, and ultimately bounded in $\Omega_{\rho_{min}}$ provided that the assumptions of Proposition 2.4 are met. □

2.3.5 Lyapunov-based MPC Using Ensemble RNN Models

In this section, we first present the formulation of Lyapunov-based model predictive control (LMPC) using a single RNN model, followed by a rigorous stability analysis for the closed-loop system. Then the LMPC using an ensemble of RNN models is presented to improve the overall predictive performance and guarantee closed-loop stability.

2.3.5.1 LMPC using a single RNN model

The design of LMPC that uses a single RNN model for prediction is given by the following optimization problem:

$$\min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), u(t)) dt \tag{2.40a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), u(t)) \tag{2.40b}$$

$$u(t) \in U, \ \forall \ t \in [t_k, t_{k+N}) \tag{2.40c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{2.40d}$$

$$\dot{\hat{V}}(x(t_k), u) \leq \dot{\hat{V}}(x(t_k), \Phi_{nn}(x(t_k))), \text{ if } x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{nn}} \tag{2.40e}$$

$$\hat{V}(\tilde{x}(t)) \leq \rho_{nn}, \ \forall \ t \in [t_k, t_{k+N}), \text{ if } x(t_k) \in \Omega_{\rho_{nn}} \tag{2.40f}$$

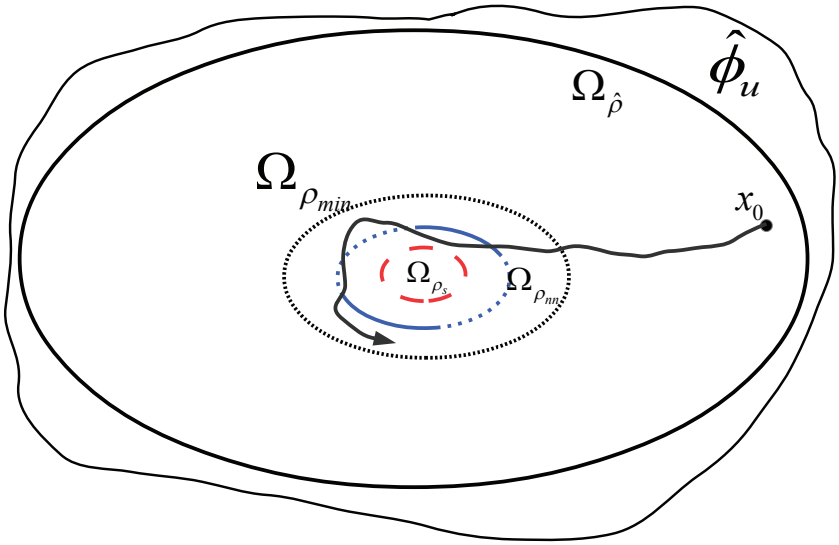


Figure 2.6: A schematic representing the set $\hat{\phi}_u$, the closed-loop stability region $\Omega_{\hat{\rho}}$, and the sets $\Omega_{\rho_{min}}$, $\Omega_{\rho_{nn}}$, Ω_{ρ_s} (going from outside to inside). Under the LMPC of Eq. 2.40, the closed-loop state is driven towards the origin and ultimately bounded in $\Omega_{\rho_{min}}$ for any $x_0 \in \Omega_{\hat{\rho}}$.

where $\tilde{x}$ is the predicted state trajectory, $S(\Delta)$ is the set of piecewise constant functions with period Δ, and N is the number of sampling periods in the prediction horizon. $\dot{\hat{V}}(x,u)$ is used to represent $\frac{\partial \hat{V}(x)}{\partial x}(F_{nn}(x,u))$. The optimal input trajectory computed by the LMPC is denoted by $u^*(t)$, which is calculated over the entire prediction horizon $t \in [t_k, t_{k+N})$. The control action computed for the first sampling period of the prediction horizon $u^*(t_k)$ is sent by the LMPC to be applied over the first sampling period and the LMPC is resolved at the next sampling time.

In the optimization problem of Eq. 2.40, the objective function of Eq. 2.40a is the integral of $L(\tilde{x}(t), u(t))$ over the prediction horizon. The constraint of Eq. 2.40b is the RNN model of Eq. 2.4 that is used to predict the states of the closed-loop system. Eq. 2.40c defines the input constraints applied over the entire prediction horizon. Eq. 2.40d defines the initial condition $\tilde{x}(t_k)$ of Eq. 2.40b, which is the state measurement at $t = t_k$. The constraint of Eq. 2.40e forces the closed-loop state to move towards the origin if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{nn}}$. However, if $x(t_k)$ enters $\Omega_{\rho_{nn}}$, the states predicted by the RNN model of Eq. 2.40b will be maintained in $\Omega_{\rho_{nn}}$ for the entire prediction horizon. A schematic of the stability region and a closed-loop state trajectory under LMPC is shown in Fig. 2.6.

Based on the LMPC of Eq. 2.40, the following theorem is established to demonstrate that the LMPC optimization problem can be solved with recursive feasibility, and closed-loop stability of the nonlinear system of Eq. 2.1 is guaranteed under the sample-and-hold implementation of the optimal control actions calculated by LMPC.

Theorem 2.2. *Consider the closed-loop system of Eq. 2.1 under the LMPC of Eq. 2.40 based on the controller $\Phi_{nn}(x)$ that satisfies Eq. 2.20. Let $\Delta > 0$, $\varepsilon_s > 0$ and $\hat{\rho} > \rho_{min} > \rho_{nn} > \rho_s$ satisfy Eq. 2.30 and 2.31. Then, given any initial state $x_0 \in \Omega_{\hat{\rho}}$, if the conditions of Proposition 2.3 and Proposition 2.4 are satisfied, there always exists a feasible solution for the optimization problem of Eq. 2.40. Additionally, it is guaranteed that under the LMPC of Eq. 2.40, $x(t) \in \Omega_{\hat{\rho}}$, $\forall t \geq 0$, and $x(t)$ ultimately converges to $\Omega_{\rho_{min}}$ for the closed-loop system of Eq. 2.1.*

Proof. We first prove that the optimization problem of Eq. 2.40 is recursively feasible for all $x \in \Omega_{\hat{\rho}}$. Specifically, if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{nn}}$ at $t = t_k$, the control action $u(t) = \Phi_{nn}(x(t_k)) \in U$, $t = [t_k, t_{k+1})$ calculated based on the state measurement $x(t_k)$ satisfies the input constraint of Eq. 2.40c and the Lyapunov-based constraint of Eq. 2.40e. Additionally, if $x(t_k) \in \Omega_{\rho_{nn}}$, the control actions given by $\Phi_{nn}(x(t_{k+i}))$, $i = 0, 1, ..., N-1$ satisfies the input constraint of Eq. 2.40c and the Lyapunov-based constraint of Eq. 2.40f since it is shown in Proposition 2.4 that the states predicted by the RNN model of Eq. 2.40b remain inside $\Omega_{\rho_{nn}}$ under the controller $\Phi_{nn}(x)$. Therefore, for all $x_0 \in \Omega_{\hat{\rho}}$, the LMPC optimization problem of Eq. 2.40 can be solved with recursive feasibility if $x(t) \in \Omega_{\hat{\rho}}$ for all times.

Next, we prove that given any $x_0 \in \Omega_{\hat{\rho}}$, the state of the closed-loop system of Eq. 2.1 is bounded in $\Omega_{\hat{\rho}}$ for all times and ultimately converges to a small neighborhood around the origin $\Omega_{\rho_{min}}$ defined by Eq. 2.31b under the LMPC of Eq. 2.40. Consider $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{nn}}$ at $t = t_k$. The constraint of Eq. 2.40e is activated such that the control action u is calculated to decrease the value of $\hat{V}(\hat{x})$ based on the states predicted by the RNN model of Eq. 2.40b over the next sampling period. Additionally, it is shown in Eq. 2.39 that if the constraint of Eq. 2.40e is satisfied, $\dot{\hat{V}}(x) \leq -\varepsilon_w$ holds for $t \in [t_k, t_{k+1})$ after the control action $u^*(t_k)$ is applied to the nonlinear system of Eq. 2.1. Therefore, the value of $\hat{V}(x)$ based on the state of the actual nonlinear system of Eq. 2.1 decreases

within the next sampling period, which implies that the closed-loop state can be driven into $\Omega_{\rho_{nn}}$ within finite sampling steps. After the state enters $\Omega_{\rho_{nn}}$, the constraint of Eq. 2.40f is activated to maintain the predicted states of the RNN model of Eq. 2.40b in $\Omega_{\rho_{nn}}$ over the entire prediction horizon. Since there exists mismatch between the RNN system of Eq. 2.40b and the nonlinear system of Eq. 2.1, the state of the nonlinear system of Eq. 2.1 may leave $\Omega_{\rho_{nn}}$ under the constraint of Eq. 2.40f. However, if we characterize a region $\Omega_{\rho_{min}}$ that satisfies Eq. 2.31b, it is shown in Proposition 2.4 that the state $x(t)$ of the nonlinear system of Eq. 2.1, $\forall t \in [t_k, t_{k+1})$ is guaranteed to be bounded in $\Omega_{\rho_{min}}$ if the predicted state by the RNN model of Eq. 2.40b remains in $\Omega_{\rho_{nn}}$. Therefore, at the next sampling step $t = t_{k+1}$, if the state $x(t_{k+1})$ is still bounded in $\Omega_{\rho_{nn}}$, the constraint of Eq. 2.40f maintains the predicted state $\hat{x}$ of the RNN model of Eq. 2.40b in $\Omega_{\rho_{nn}}$ such that the actual state x of the nonlinear system of Eq. 2.1 stays inside $\Omega_{\rho_{min}}$. However, if $x(t_{k+1}) \in \Omega_{\rho_{min}} \backslash \Omega_{\rho_{nn}}$, following the proof we have shown for the case that $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{nn}}$, the constraint of Eq. 2.40e will be activated instead to drive it towards the origin. This completes the proof of boundedness of the states of the closed-loop system of Eq. 2.1 in $\Omega_{\hat{\rho}}$ and convergence to $\Omega_{\rho_{min}}$ for any $x_0 \in \Omega_{\hat{\rho}}$. □

Remark 2.8. *Theorem 2.2 shows that closed-loop stability of the nonlinear system of Eq. 2.1 is achieved under the LMPC of Eq. 2.40 that is designed based on the RNN model of Eq. 2.4 and RNN-based constraints. It is noted that the closed-loop state of the nonlinear system of Eq. 2.1 can be driven to a small neighborhood around the origin because the constraints of the LMPC of Eq. 2.40 guarantee the decrease of $\hat{V}$ in each sampling period accounting for the effect of model mismatch including the modeling error ν between the system of Eq. 2.1 and the RNN model of Eq. 2.4, the sample-and-hold implementation of control actions, and the bounded disturbances $w(t)$ in Eq. 2.1. In other words, closed-loop stability can be maintained under the LMPC of Eq. 2.40 if the modeling error ν, the sampling period Δ and the bound of disturbances w_m are sufficiently small such that Proposition 2.3 and Proposition 2.4 are satisfied.*

2.3.5.2 LMPC using an ensemble of RNN models

Since the RNN model accuracy plays an important role in the optimization problem of LMPC of Eq. 2.40, ensemble regression models introduced in previous section are employed to improve the performance of the closed-loop system of Eq. 2.1 under LMPC. Based on the formulation of LMPC given by Eq. 2.40, the LMPC that incorporates ensemble regression models are developed as follows:

$$\min_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t),u(t))dt \tag{2.41a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = \frac{1}{N_e}\sum_{j=1}^{N_e} F_{nn}^{j}(\tilde{x}(t),u(t)) \tag{2.41b}$$

$$u(t) \in U,\ \forall\ t \in [t_k,t_{k+N}) \tag{2.41c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{2.41d}$$

$$\dot{\hat{V}}(x(t_k),u) \leq \dot{\hat{V}}(x(t_k),\Phi_{nn}(x(t_k))),\ \text{if}\ x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{nn}} \tag{2.41e}$$

$$\hat{V}(\tilde{x}(t)) \leq \rho_{nn},\ \forall\ t \in [t_k,t_{k+N}),\ \text{if}\ x(t_k) \in \Omega_{\rho_{nn}} \tag{2.41f}$$

where the notation follows that in Eq. 2.40 and N_e is the number of regression models used for prediction. N_e can be the number of all available RNN models, or can be determined off-line through trial-and-error to account for computational efficiency. It is shown in Eq. 2.41b that the states $\tilde{x}(t)$, $t \in [t_k,t_{k+N})$ are now predicted by taking the average of RNN models F_{nn}^{j}, $j = 1,...,N_e$. Since the objective function of Eq. 2.41a and the Lyapunov-based constraints of Eq. 2.41e-2.41e are computed based on predicted states from Eq. 2.41b, the application of ensemble regression models in Eq. 2.41b significantly improves the solution of the optimization problem and thus leads to a better closed-loop performance. Additionally, it is readily shown that closed-loop stability established in Theorem 2.2 is still guaranteed for the LMPC of Eq. 2.41 because each regression model F_{nn}^{j}, $j = 1,...,N_e$ is trained to satisfy all the conditions and assumptions in Theorem 2.2.

The LMPC based on ensemble regression models is implemented in the same way as the LMPC of Eq. 2.40, i.e., the optimal input trajectory $u^*(t)$ is calculated over the entire prediction horizon

$t \in [t_k, t_{k+N})$ but only the first control action $u^*(t_k)$ is applied to the system of Eq. 2.1 over the first sampling period. However, since the optimization problem of the LMPC of Eq. 2.41 is now based on prediction results from multiple RNN models, the computation time for training multiple RNN models and solving the LMPC of Eq. 2.41 both increase rapidly as the number of RNN models being used increases, which suggests the further investigation on computational efficiency for the real-time implementation of LMPC using an ensemble regression models.

2.3.6 Parallel Computing

An ensemble of RNN models has been utilized in LMPC to provide more accurate prediction of future states through the average of multiple RNN prediction results. As a result, the closed-loop performance can be improved in the sense that the closed-loop state of the system of Eq. 2.1 is able to converge to the origin quickly and smoothly. While ensemble learning improves model prediction accuracy by using multiple RNN models, computation time for running LMPC is inevitably increased. Therefore, considering the significant increase of computation time arising from the use of multiple RNN models, parallel computing is employed to reduce real-time computation time. Parallel computing is a type of computation in which the execution of multiple processes is carried out simultaneously [10]. Generally, it takes advantage of multiple compute resources (e.g., a single computer with multiple processors/cores or many computers connected by a network) to solve a computationally heavy task, in which a complex problem can be broken into discrete parts that can be solved concurrently. Additionally, parallel computing can be categorized into two types based on whether there exists communication between processors/networked computers: 1) In parallel computing without communication, multiple processors execute multiple tasks simultaneously and generate the results independently. 2) In parallel computing with communication (sometimes it is also called distributed computing), networked computers or multiple processors communicate and coordinate the work through message passing interface (MPI) to obtain final results. Based on the computation tasks for training multiple RNN models and calculating the average of multiple RNN prediction results in LMPC, the first type and the second

type of parallel computing are applied to these two tasks, respectively, to enhance computational efficiency in both cases.

2.3.6.1 Training multiple RNNs in parallel

Multiple RNN models are constructed via a k-fold cross validation method discussed in Section 2.2.3. Specifically, if k RNN models are utilized in the LMPC of Eq. 2.41, the computation time for training all RNN models in series is approximately k times longer than that for a single RNN model. It is noted that the resulting increase of computation time is unnecessary since the training processes for k RNN models are independent from each other. Therefore, parallel computing is utilized to distribute the training processes to multiple processors such that k RNN models can be trained simultaneously. The training processes of k RNN models are implemented in parallel with the following steps: 1) k processors are first reserved with sufficient memory. 2) Based on k-fold cross validation, the entire dataset is partitioned into k folds with the same size, which are then distributed to all reserved processors. 3) For the kth processor, the RNN model is trained with $k-1$ subsets (i.e., the kth subset is excluded) as the training dataset and the remaining kth subset as the validation dataset. 4) A bash script is created to run all k processors together such that the training processes for the k RNN models can be executed concurrently. Since the stopping criteria might not be satisfied by the k training processes simultaneously due to different training datasets, the total computation time is determined by the slowest training process.

2.3.6.2 Parallel operation of LMPC using an ensemble of RNNs

An ensemble of RNN models is utilized in the LMPC of Eq. 2.41, under which prediction accuracy is improved and closed-loop stability of the nonlinear system remains valid. Since the optimal solution $u^*(t)$ is now computed based on the states predicted by multiple RNN models, the computation time for an ensemble of N_e RNN models increases rapidly (at least N_e times the original computation time for the LMPC based on a single RNN model) under serial computation of Eq. 2.41b, which greatly limits the application of ensemble regression model-based LMPC in

industry. Therefore, in this subsection, parallel computing is utilized to reduce the computation time of calculating multiple RNN models of Eq. 2.41b.

Specifically, in the LMPC optimization problem of Eq. 2.41, the state prediction given by Eq. 2.41b can be broken apart into N_e similar sub-tasks that can be processed independently and simultaneously. Consider using N_e ($N_e \leq k$) RNN models for prediction of Eq. 2.41b. The calculation of Eq. 2.41b through parallel computing consists of the following steps: 1) As shown in Fig. 2.7, we first reserve $N_e + 1$ nodes, in which node 0 is the host node and the rest are worker nodes. The host node is used to receive and send information while the worker nodes are mainly used for computation. 2) The optimization problem is running on the host node while the computation of multiple RNN models is assigned to worker nodes. Specifically, when it comes to state prediction using Eq. 2.41b, the host node is executed first to broadcast $x(t_k)$ and $u(t)$ to all nodes since ensemble regression models in Eq. 2.41b share the same initial condition $x(t_k)$ and the same guess of control actions $u(t)$ at $t = t_k$. 3) Each worker node is assigned with an RNN model for prediction and the host node gathers the results from worker nodes and compute the average as the final result. 4) The optimal control action $u^*(t_k)$ is sent to the real system to be applied for the next sampling period by the host node. The above process is repeated every sampling step (i.e., at the next sampling time t_{k+1}, the LMPC of Eq. 2.41 receives the state measurement $x(t_{k+1})$ and sends it to the host node. Then, steps 1-4 are repeated to parallelize the computation of Eq. 2.41b.)

Remark 2.9. *Computational efficiency of the LMPC optimization problem of Eq. 2.41 is significantly improved through the parallel operation of N_e independent ensemble regression models. However, it is noted that the computation time may not be reduced exactly by N_e times under parallel operation due to the communication and waiting time between the host node and the worker nodes. It is also important to mention that the communication between the LMPC and the process model, and the main program of the optimization problem itself are running on the host node only. Additionally, as shown in Fig. 2.7, synchronization operation should be employed when the host node combines all the results from worker nodes to ensure that each task in worker node blocks until all tasks in the computing group reach the host node.*

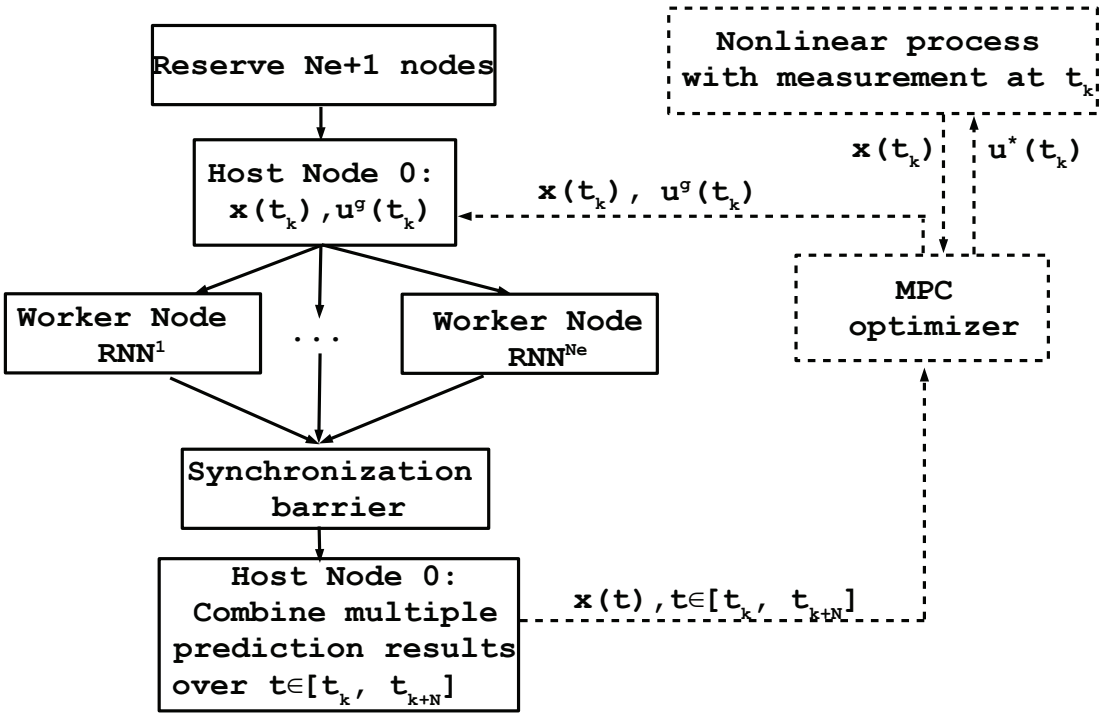


Figure 2.7: Parallel computation of the ensemble of RNN models in LMPC, where $u^g(t_k)$ represents the guess of control action sent to the RNN models.

2.3.7 Computational Implementation Issues of RNN Models

In this section, we address computational implementation issues for the RNN models obtained following the training algorithm in Section 2.2. Specifically, the implementation of RNN models for long prediction horizon is first discussed. Then, numerical methods are employed to evaluate modeling error and approximate the Lyapunov-based constraints in LMPC, respectively.

2.3.7.1 Long prediction horizon

Although the ensemble of RNN models developed in Section 2.2 is to predict future states over $t \in [t_k, t_k + P_{nn}]$ given the states and inputs at $t = t_k$, where P_{NN} is an integer multiple of the sampling period Δ, it is noted that ensemble regression models can be applied to predict states for longer period of time (i.e., $t \in [t_k, t_k + NP_{nn}]$, $N > 1$) in practical applications, e.g., model predictive control. Specifically, the obtained RNN models will be utilized successively at every prediction step $t = t_k + iP_{nn}$, $i = 0, 1, ..., N-1$, to predict all the states within the entire prediction horizon $t \in [t_k, t_k + NP_{nn}]$, in which the prediction results (i.e., the output vector $x(t_k + iP_{nn})$) from the previous RNN models will be used as the initial states for the current prediction to predict states over $[t_k + iP_{nn}, t_k + (i+1)P_{nn}]$, $i = 0, 1, ..., N-1$. Additionally, since the means and the standard deviations for normalizing inputs and re-scaling outputs could be slightly different, intermediate re-scaling and normalizing steps should be performed between two successive ensemble prediction steps during the entire prediction horizon.

Before we apply the obtained RNN models within LMPC, the testing dataset that has not been used in the training process is utilized to test the prediction performance of RNNs. In this case, the normalizing and re-scaling functions before and after the ensemble of RNN models (Fig. 2.3) should be updated with the statistics of the testing dataset. Specifically, the normalizing and re-scaling functions during the training process are constructed based on the statistics of the training dataset only instead of the entire dataset due to the following reasons. First, the training and testing datasets may not be equally representative of the operating region considered, and thus, the training and testing datasets should be normalized separately. Second, data leakage that

introduces information from outside, e.g., testing dataset, into RNN model should be prevented during the training process to avoid creating an overly optimistic but potentially invalid predictive model. Therefore, based on the normalizing and re-scaling functions designed for the testing dataset, the prediction performance of RNN models is evaluated by the mean absolute percentage error between the predicted states of the RNN models and the actual states derived from the nominal nonlinear system $\dot{x} = f(x) + g(x)u$.

Remark 2.10. *While the use a longer prediction horizon by recursively performing RNN predictions in LMPC can improve the closed-loop performance, a short horizon may be computationally advantageous for real-time application. Also, it should be noted that closed-loop stability properties derived in the previous sections hold for any prediction horizon size. Therefore, the length of the prediction horizon should be determined via closed-loop simulations to balance optimality of the LMPC solutions and its computational complexity.*

2.3.7.2 Approximation via numerical methods

Since we mainly discuss the continuous RNN models in Section 2.2, while in practice, the datasets for training RNN models are mostly generated by a sample-data collection from industrial processes, lab experiments or numerical simulation, necessary approximations should be performed to incorporate the RNN model trained on sample data within LMPC. Specifically, numerical methods are utilized to compute modeling error, characterize the closed-loop stability region Ω_ρ for the RNN model and calculate $\dot{V}(x(t_k), u(t_k))$ in the LMPC constraint of Eq. 2.41e, respectively.

a) Approximation of modeling error

Since the RNN is trained to predict future states over $t \in [t_k, t_k + P_{nn})$, in which the RNN output is the state at $t_k + P_{nn}$ and the time interval between internal states is chosen as the integration time step h_c, the modeling error $\nu = \dot{x}(t_k) - \dot{\hat{x}}(t_k)$ at the state $x(t_k) = \hat{x}(t_k)$ is approximated using a

forward finite difference method during the training process as follows:

$$
\begin{aligned}
|\nu| &= \left| \frac{x(t_k + h_c) - x(t_k)}{h_c} - \frac{\hat{x}(t_k + h_c) - \hat{x}(t_k)}{h_c} \right| \\
&= \left| \frac{x(t_k + h_c) - \hat{x}(t_k + h_c)}{h_c} \right|
\end{aligned}
\tag{2.42}
$$

where h_c is a sufficiently small time interval. $x(t_k + h_c)$ is obtained via explicit Euler method with an integration time step h_c, and $\hat{x}(t_k + h_c)$ is the first internal state of the RNN model. Then, the constraint $|\nu| \leq \gamma|x|$ is satisfied if the following equation holds:

$$
\left| \frac{x(t_k + h_c) - \hat{x}(t_k + h_c)}{x(t_k + h_c)} \right| \leq \gamma h_c \tag{2.43}
$$

According to Eq. 2.43, the mean absolute percentage error between predicted states $\hat{x}$ and targeted states x in training data can be utilized as a metric to indicate the modeling error of RNNs.

b) Characterization of closed-loop operating region

The stabilizing controller $u = \Phi_{nn}(x) \in U$ is initially utilized to characterize the set ϕ_u and the closed-loop stability region Ω_ρ based on the RNN model written in the form of $\dot{\hat{x}} = \hat{f}(\hat{x}) + \hat{g}(\hat{x})u$. However, since it is difficult to derive the explicit forms of $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ for an RNN with a complex structure, numerical methods are utilized to approximate $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$. For example, $\hat{f}(\cdot)$ can be approximated by the predicted $\dot{\hat{x}}$ with $u = 0$, where $\dot{\hat{x}}$ is obtained using the forward finite difference method as shown in the previous section. Then, $\hat{g}(\cdot)$ is approximated by $\hat{g}(\hat{x}) = (\dot{\hat{x}} - \hat{f}(\hat{x}))/u$ with a nonzero u. Since the minimum prediction step in RNNs is the sufficiently small integration time step h_c, the approximation results via numerical methods can be regarded as a good representation of the actual $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ of an RNN model. After $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ are obtained, a simulation with a full sweep over the entire state-space based on the stabilizing controller $u = \Phi_{nn}(x) \in U$ is performed to characterize the region ϕ_u and $\dot{V}(x) = \frac{\partial V(x)}{\partial x}(F_{nn}(x,u))$ is approximated via forward finite difference method. Subsequently, the closed-loop stability region Ω_ρ is characterized as a level set of $V(x)$.

c) Approximation of Lyapunov-based constraints

Additionally, $\dot{V}(x(t_k),u(t_k))$ in the Lyapunov-based constraint of Eq. 2.41e is approximated via the same numerical method (i.e., forward finite difference method). It is noted that the approximation of $\dot{V}(x(t_k),u(t_k))$ does not affect closed-loop stability of the actual nonlinear system (i.e., $\dot{x} = F(x,u,w) := f(x) + g(x)u + h(x)w$) under the constraint of Eq. 2.41e since the same numerical method is used to approximate both $\dot{V}(x(t_k),u(t_k))$ and $\dot{V}(x(t_k),\Phi_{nn}(x))$. Specifically, it has been shown that the controller $u = \Phi_{nn}(x) \in U$ is able to stabilize the actual nonlinear system at the origin for all x in Ω_ρ since Eq. 2.2 is satisfied in $\Omega_\rho \subset \phi_u$ that is characterized via the numerical computation of $\dot{V}(x(t_k),\Phi_{nn}(x))$.

2.3.8 Application to a Chemical Process Example

A chemical process example is used to illustrate the application of LMPC using RNN models to maintain the closed-loop state within the stability region. Specifically, a well-mixed, non-isothermal continuous stirred tank reactor (CSTR) where an irreversible second-order exothermic reaction takes place is considered. The reaction transforms a reactant A to a product B ($A \rightarrow B$). The inlet concentration of A, the inlet temperature and feed volumetric flow rate of the reactor are C_{A0}, T_0 and F, respectively. The CSTR is equipped with a heating jacket that supplies/removes heat at a rate Q. The CSTR dynamic model is described by the following material and energy balance equations:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 \tag{2.44a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{2.44b}$$

where C_A is the concentration of reactant A in the reactor, V is the volume of the reacting liquid in the reactor, T is the temperature of the reactor and Q denotes the heat input rate. The concentration of reactant A in the feed is C_{A0}. The feed temperature and volumetric flow rate are T_0 and F, respectively. The reacting liquid has a constant density of ρ_L and a heat capacity of C_p. ΔH, k_0, E,

and R represent the enthalpy of reaction, pre-exponential constant, activation energy, and ideal gas constant, respectively. Process parameter values are listed in Table 2.1.

Table 2.1: Parameter values of the chemical reactor example.

$T_0 = 300\ K$	$F = 5\ m^3/hr$
$V = 1\ m^3$	$E = 5 \times 10^4\ kJ/kmol$
$k_0 = 8.46 \times 10^6\ m^3/kmol\ hr$	$\Delta H = -1.15 \times 10^4\ kJ/kmol$
$C_p = 0.231\ kJ/kg\ K$	$R = 8.314\ kJ/kmol\ K$
$\rho_L = 1000\ kg/m^3$	$C_{A0_s} = 4\ kmol/m^3$
$Q_s = 0.0\ kJ/hr$	$C_{A_s} = 1.22\ kmol/m^3$
$T_s = 438\ K$	

The CSTR is initially operated at the unstable steady-state $(C_{As},\ T_s) = (1.95\ kmol/m^3,\ 402\ K)$, and $(C_{A0_s}\ Q_s) = (4\ kmol/m^3,\ 0\ kJ/hr)$. The manipulated inputs are the inlet concentration of species A and the heat input rate, which are represented by the deviation variables $\Delta C_{A0} = C_{A0} - C_{A0_s}$, $\Delta Q = Q - Q_s$, respectively. The manipulated inputs are bounded as follows: $|\Delta C_{A0}| \leq 3.5\ kmol/m^3$ and $|\Delta Q| \leq 5 \times 10^5\ kJ/hr$. Therefore, the states and the inputs of the closed-loop system are $x^T = [C_A - C_{As}\ T - T_s]$ and $u^T = [\Delta C_{A0}\ \Delta Q]$, respectively, such that the equilibrium point of the system is at the origin of the state-space, (i.e., $(x_s^*, u_s^*) = (0,0)$). The control objective is to operate the CSTR at the unstable equilibrium point $(C_{As},\ T_s)$ by manipulating the heat input rate ΔQ and the inlet concentration ΔC_{A0} under the LMPC using RNN models. The explicit Euler method with an integration time step of $h_c = 10^{-4}\ hr$ is applied to numerically simulate the dynamic model of Eq. 2.44. The nonlinear optimization problem of the LMPC of Eq. 2.41 is solved using the python module of the IPOPT software package [158], named PyIpopt with the sampling period $\Delta = 10^{-2}\ hr$.

2.3.8.1 Data generation

To apply the LMPC of Eq. 2.41 to the CSTR of Eq. 2.44, extensive open-loop simulations are first conducted in the closed-loop stability region Ω_ρ for the CSTR of Eq. 2.44 to generate the dataset

for RNN models, and subsequently, RNN models are developed to capture the system dynamics in Ω_ρ with a desired degree of accuracy. The control Lyapunov function $V(x) = x^T Px$ is designed with $P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix}$. Then, the closed-loop stability region Ω_ρ for the CSTR with $\rho = 372$ is characterized as a level set of Lyapunov function inside the region ϕ_u, from which the origin can be rendered exponentially stable under the controller $u = \Phi(x) \in U$. Open-loop simulations are performed with a full sweep through all of the feasible initial conditions $x_0 \in \Omega_\rho$ and inputs $u \in U$ for finite sampling steps, from which the state trajectories represented by sampled data points are collected with a minimum time step as the integration time step h_c. Subsequently, the RNN is developed based on the dataset generated from open-loop simulations in Ω_ρ to predict future states over one sampling period Δ with the minimum prediction period h_c using the state-of-the-art API, Keras. Specifically, the RNN model is designed to have two hidden recurrent layers consisting of 96 and 64 recurrent units, respectively and use the sigmoid function as the activation function. The stopping criteria for the training process includes the modeling error less than a threshold and early stopping being triggered. Additionally, a 10-fold cross validation are used to construct homogeneous ensemble regression models for the LMPC of Eq. 2.41 using multiple RNN models. After the RNN model is obtained, the Lyapunov function $\hat{V}(x)$ for the RNN model is chosen to be the same as $V(x)$, and the set $\hat{\phi}_u$, in which $\dot{\hat{V}} \leq k\hat{V}$ holds, is characterized in Fig. 2.8 using the controller $u = \Phi_{nn}(x) \in U$ with the approximation approach discussed before. The closed-loop stability region $\Omega_{\hat{\rho}}$ for the CSTR system described by the RNN model is characterized as the largest level set of $\hat{V}$ in $\hat{\phi}_u$ and also a subset of Ω_ρ (i.e., $\Omega_{\hat{\rho}} \subset \Omega_\rho$) with $\hat{\rho} = 368$. Additionally, $\rho_{nn} = 1.6$ and $\rho_{min} = 2$ are determined through extensive simulations for $u \in U$. The LMPC cost function of Eq. 2.41a is designed to be $L(x,u) = |x|^2_{Q_1} + |u|^2_{Q_2}$, where $Q_1 = [500 \;\; 0; \; 0 \;\; 0.5]$ and $Q_2 = [1 \;\; 0; \; 0 \;\; 8 \times 10^{-11}]$, such that the minimum value of L is achieved at the origin. It is noted that since the steady-state $(C_{As}, \; T_s) = (1.95 \; kmol/m^3, \; 402 \; K)$ is an unstable equilibrium point of the system of Eq. 2.44, open-loop simulations are performed for a few sampling periods only to guarantee that state trajectories starting from Ω_ρ do not diverge quickly and can be bounded in a

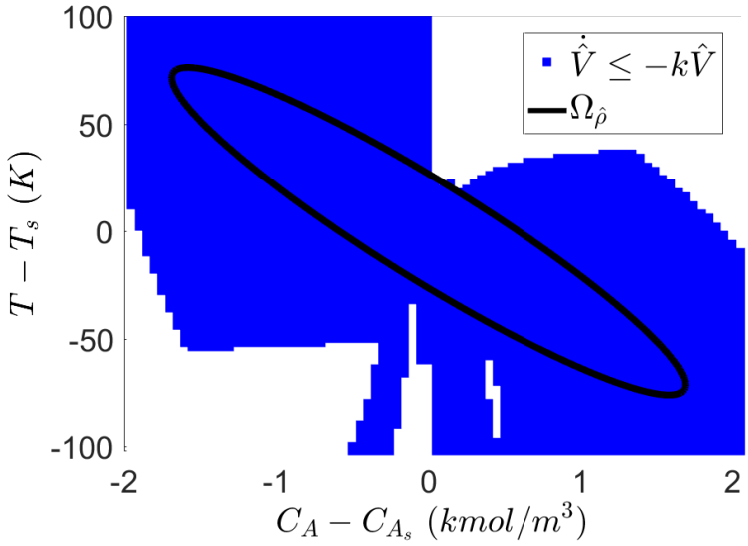


Figure 2.8: The set $\hat{\phi}_u$ represented by the blue region and the stability region $\Omega_{\hat{\rho}}$ (black ellipse) for the closed-loop CSTR under the controller $u = \Phi_{nn}(x) \in U$.

slightly larger region.

2.3.8.2 Linear state-space model

To illustrate the effectiveness of the proposed LMPC of Eq. 2.41 using RNN models, we also compare it with the LMPC using a linear state-space model and the first-principles model of Eq. 2.44, respectively. The linear state-space model for the CSTR system of Eq. 2.44 is identified with the following form:

$$\dot{x} = A_s x + B_s u \tag{2.45}$$

where x and u are the state vector and the manipulated input vector, A_s and B_s are coefficient matrices for the state-space model. Following the system identification method in [74], the numerical algorithms for subspace state space system identification is utilized to obtain A_s and B_s as $A_s = 100 \times \begin{bmatrix} -0.154 & -0.003 \\ 5.19 & 0.138 \end{bmatrix}$ and $B_s = \begin{bmatrix} 4.03 & 0 \\ 1.23 & 0.004 \end{bmatrix}$. The eigenvalues of matrix A_s is calculated to be $\lambda_1 = -5$ and $\lambda_2 = 3.14$, which is consistent with the fact that the steady-state $(C_{As},\ T_s) = (1.95\ kmol/m^3,\ 402\ K)$ is an unstable equilibrium point of CSTR.

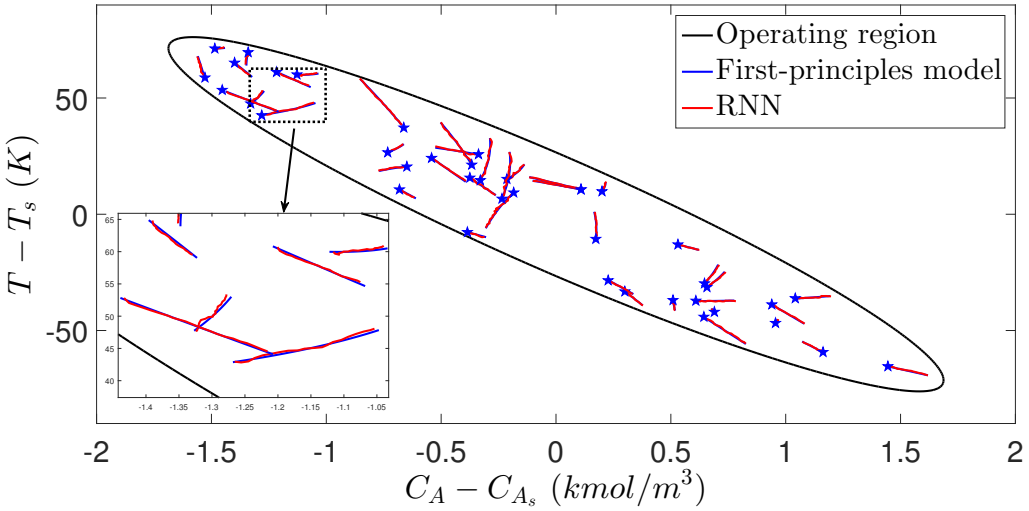


Figure 2.9: The state-space profiles for the open-loop simulation using the first-principles model of Eq. 2.44 and the RNN model, respectively, for various sets of inputs and initial conditions (marked as blue stars) x_0 in the closed-loop stability region $\Omega_{\hat{\rho}}$.

2.3.8.3 Simulation results

We first carry out simulation results under the LMPC using the RNN model and the first-principles model of Eq. 2.44, respectively. It should be noted that the machine learning approach is used when only data are available. The first-principles model in the following simulations substitutes for the role of the experimental/industrial process. In other words, the MPC using first-principles model only serves as a benchmark to determine the best performance that any data-driven modeling method can achieve. In Fig. 2.9, it is demonstrated that starting from the same initial condition $x_0 \in \Omega_{\hat{\rho}}$ with the same input sequences, the state trajectories for a fixed finite interval of time under the RNN model are close to those under the first-principles model of the nonlinear CSTR of Eq. 2.44. This implies that the well-trained RNN model can be regarded as a good representation for the CSTR first-principles model of Eq. 2.44. Next, the RNN model is incorporated in the LMPC of Eq. 2.41 using a single RNN model, under which the closed-loop state trajectories, state and manipulated input profiles of the system of Eq. 2.44 are shown in Figs. 2.10-2.13.

Fig. 2.10 demonstrates that for initial conditions $x_0 \in \Omega_{\hat{\rho}}$, the closed-loop state is bounded in the closed-loop stability region $\Omega_{\hat{\rho}}$ for all times and ultimately converges to a small neighborhood

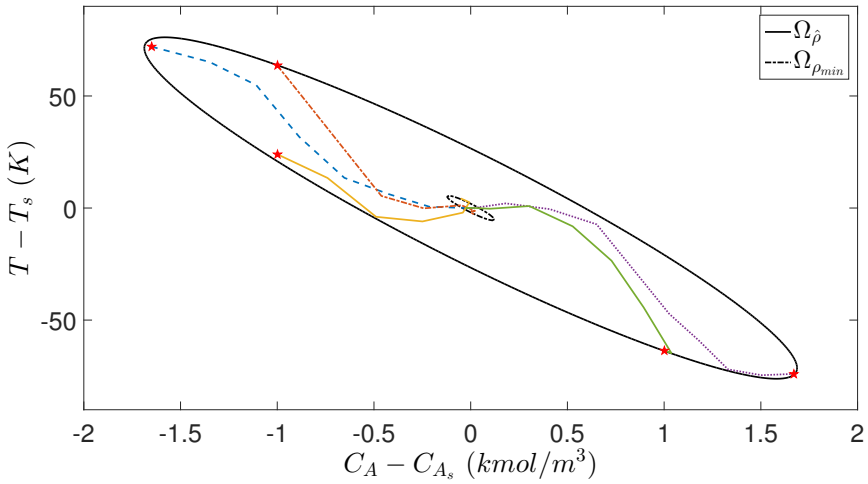


Figure 2.10: The state-space profiles for the closed-loop CSTR under the LMPC of Eq. 2.41 using RNN models for various initial conditions (marked as red stars) in the closed-loop stability region $\Omega_{\hat{\rho}}$.

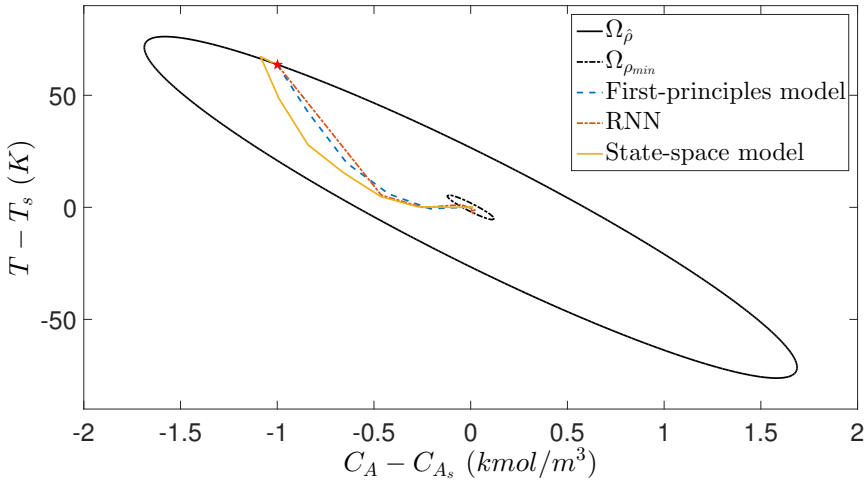


Figure 2.11: The state-space profiles for the closed-loop CSTR under the LMPC using the following models: the first-principles model (blue trajectory), the RNN model (red trajectory) and the linear state-space model (yellow trajectory) for an initial condition (-1, 63.6).

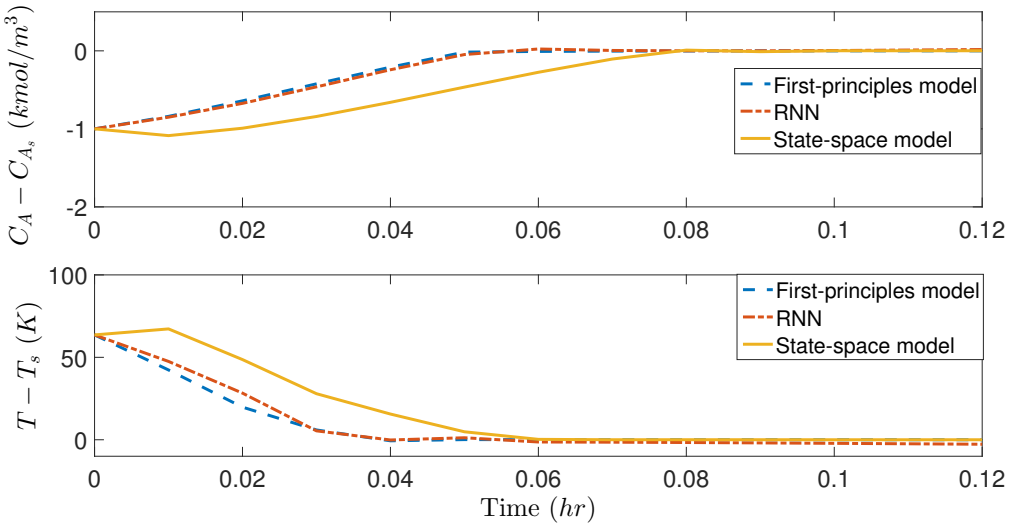


Figure 2.12: The state profiles ($x_1 = C_A - C_{As}$ and $x_2 = T - T_s$) for the initial condition (-1, 63.6) under the LMPC using the following models: the first-principles model (blue trajectory), the RNN model (red trajectory) and the linear state-space model (yellow trajectory).

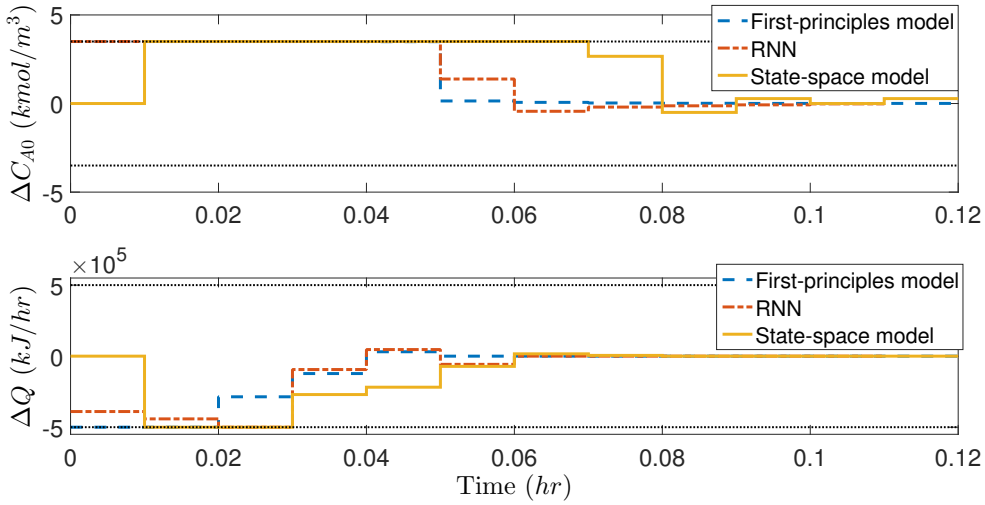


Figure 2.13: Manipulated input profiles ($u_1 = \Delta C_{A0}$ and $u_2 = \Delta Q$) for the initial condition (-1, 63.6) under the LMPC using the following models: the first-principles model (blue trajectory), the RNN model (red trajectory) and the linear state-space model (yellow trajectory), where the black dotted lines represent the upper and lower bound for u_1 and u_2, respectively.

around the origin ($\Omega_{\rho_{min}}$) under the LMPC of Eq. 2.41 using a single RNN model. Additionally, Fig. 2.11 shows the comparison of state trajectories for the closed-loop system under the LMPC using a single RNN model, the state-space model of Eq. 2.45 and the first-principles model of Eq. 2.44, respectively. It is demonstrated that in all cases, the state of the closed-loop system of Eq. 2.44 is maintained within $\Omega_{\hat{\rho}}$ for all times and driven to $\Omega_{\rho_{min}}$ under LMPC for an initial condition $x_0 = (-1, 63.6)$. However, through the comparison of state profiles under the LMPC using three different models in Figs. 2.12, it is shown that the state trajectory under the RNN model stays closer to the one under the actual nonlinear model of Eq. 2.44, and thus, takes less time to settle to the steady-state compared to the LMPC using the state-space model. It is also noted that although the LMPC using the state-space model performs well for some initial conditions close to the origin, it shows oscillation for initial conditions near the boundary of the closed-loop stability region $\Omega_{\hat{\rho}}$ because the linear state-space model of Eq. 2.45 is not able to capture the nonlinearities of the CSTR in this region. Therefore, the LMPC using RNN model outperforms the one using state-space model in terms of faster convergence speed and improved closed-loop stability. Fig. 2.13 depicts the manipulated input profiles in deviation from the steady-state values, where the dashed horizontal lines are the upper and lower bounds for the manipulated inputs. It is shown that the input constraints are met for all times under the LMPC of Eq. 2.41 using all three models.

2.3.8.4 Parallel computation of ensemble regression models

So far, we have demonstrated that the LMPC with a single RNN model is able to drive the closed-loop state to $\Omega_{\rho_{min}}$, and compared the closed-loop performance of the system of Eq. 2.44 under the LMPC with the RNN models based on a large dataset, and a dataset with a lower amount of data, respectively. In this section, we apply the LMPC using ensemble regression models to the CSTR of Eq. 2.44 and perform parallel computing to improve computational efficiency. Since it is common that the RNN model may not perform perfectly for some initial conditions due to insufficient data, the utilization of ensemble regression models may improve the overall

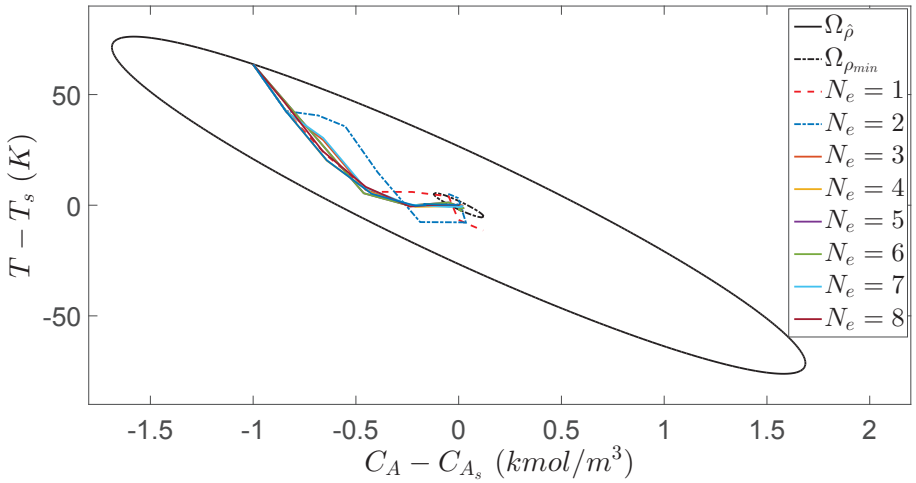


Figure 2.14: The state-space profiles for the closed-loop CSTR under the LMPC using the following models: the first-principles model (blue trajectory), the RNN model (red trajectory) and the linear state-space model (yellow trajectory) for an initial condition (-1, 63.6).

performance of RNN models within the entire operating region.

Figs. 2.14 shows the state trajectories under the LMPC using various numbers of regression models (i.e., $N_e = 1, ..., 8$). It is observed that starting from the initial condition (-1, 63.6), the closed-loop system of Eq. 2.44 does not converge to the origin smoothly using a single RNN model that is trained poorly around the origin. Additionally, it is shown that as the number of regression models used in LMPC increases, the closed-loop performance is improved in terms of less oscillation and faster convergence. Therefore, in this case, the optimal number of regression models is determined to be five as no further improvement is noticed for the increase of regression models being used.

However, as more regression models are utilized in the LMPC of Eq. 2.41, the computation time under serial operation increases significantly, which makes it challenging for the controller to be implemented in practice. Therefore, to address the computational efficiency issue, we run the LMPC of Eq. 2.41 in the parallel mode. Specifically, a Message Passing Interface (MPI) for the Python programming language, named *MPI4Py* [40], is incorporated in the program of the LMPC optimization problem to break the prediction models of Eq. 2.41b into five independent computing processes. Additionally, since the main program of the LMPC optimization problem

is executed on the host node only, we use a *while-loop* and synchronization mechanism to ensure that all worker nodes work with the host node simultaneously throughout the optimization process. The parallel computing of the LMPC optimization problem of Eq. 2.41 is solved on the UCLA Hoffman2 Distributed Cluster.

The averaged computation time for solving the LMPC optimization problem per sampling step using the first-principles model of Eq. 2.44, the linear state-space model of Eq. 2.45, a single RNN model, five ensemble regression models in serial mode, and five ensemble regression models in parallel mode are reported in Table 2.2. In Table 2.2, it is shown that the LMPC optimization problem using state-space model is solved with the shortest computation time. The optimization problem of LMPC using RNN models is time-consuming compared to the state-space model or the first-principles model due to the large number of internal states, the essential normalization and data reshaping steps, and the communication between host and worker nodes. However, it is shown that under parallel operation, the computation time for solving the LMPC optimization problem using five ensemble regression models at each sampling step is around 11 s, which is significantly reduced (approximately 5 times less than the serial computing), and becomes less than the sampling period (i.e., $\Delta = 0.01\ hr = 36\ s$). This implies that the LMPC using an ensemble of RNN models can be implemented in real-time if parallel computing is employed. Additionally, the computation time for solving LMPC under RNN models may be further reduced if TensorFlow is employed, which is more computationally efficient than Keras.

Table 2.2: Computation time for solving the LMPC using different models.

Models	Computation time
First-principles model	$<$ 1 second
Linear state-space model	$<$ 0.1 second
Single RNN model	$\sim$ 8 second
Ensemble regression models in serial mode	$>$ 50 second
Ensemble regression models in parallel mode	$\sim$ 11 second

2.3.8.5 RNN model performance evaluation

To illustrate the advantages of the ensemble of RNN models, in this section, we characterize the region of initial conditions $x_0 \in \Omega_{\hat{\rho}}$ for which the performance of the LMPC using the data-driven model (i.e., the state-space model of Eq. 2.45 and the ensemble of RNN models, respectively) is close to that of the LMPC using the first-principles model of Eq. 2.44. Specifically, extensive closed-loop simulations that sweep over all the initial conditions x_0 in the closed-loop stability region $\Omega_{\hat{\rho}}$ are conducted under the LMPC of Eq. 2.41 using the following models: the first-principles model of Eq. 2.44, the ensemble of RNN models and the state-space model of Eq. 2.45. It should be mentioned that methods that may improve the performance of linear state-space model, e.g., the ensemble of linear state-space models and multiple linear state-space models for different portions of the closed-loop stability region $\Omega_{\hat{\rho}}$, are not investigated in this work since the aim of this study is to develop a computationally efficient LMPC scheme using an ensemble of RNN models.

All the closed-loop simulations are run with a fixed time length that is sufficiently long for the closed-loop state to converge to $\Omega_{\rho_{min}}$ for any initial condition $x_0 \in \Omega_{\hat{\rho}}$. Extensive closed-loop simulations demonstrate that the LMPC using the ensemble of RNN models and the LMPC using the state-space model of Eq. 2.45 both drive the closed-loop state to $\Omega_{\rho_{min}}$ for any initial condition $x_0 \in \Omega_{\hat{\rho}}$. Therefore, to compare the performance of closed-loop system under different data-driven models, a performance index S is introduced to calculate the relative error between the closed-loop states under the data-driven model and the first-principles model as follows:

$$S = \frac{\sum_{i=1}^{L} |\hat{V}(x_i^d) - \hat{V}(x_i^f)|}{\sum_{i=1}^{L} \hat{V}(x_i^f)} \tag{2.46}$$

where L is number of sampling steps in simulation, x_i^f represents the ith closed-loop state for the first-principles model of Eq. 2.44, and x_i^d represents the ith closed-loop state for the data-driven models, which are the ensemble of RNN models and the linear state-space model of Eq. 2.45, respectively. Since the value of $\hat{V}(x)$ decreases as the state moves towards the origin under

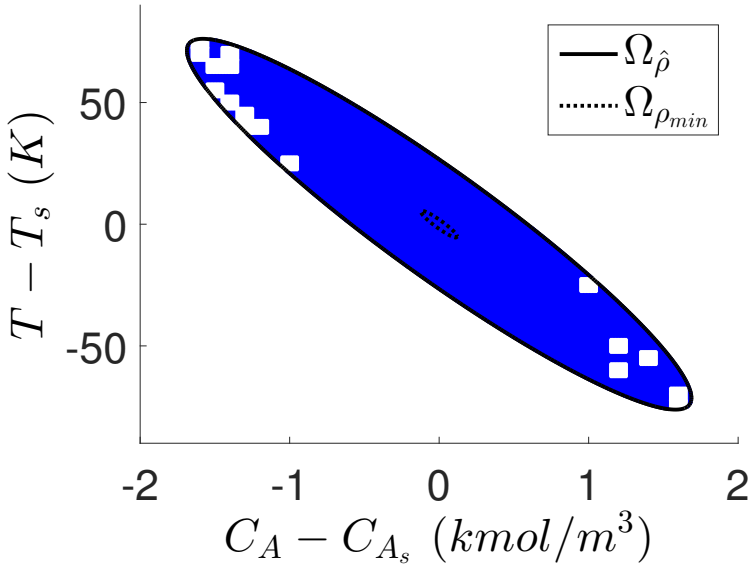


Figure 2.15: The set of initial conditions $x_0 \in \Omega_{\hat{\rho}}$ (marked as blue points) in which the closed-loop CSTR under the LMPC using the linear state-space of Eq. 2.45 behaves similarly to the LMPC using the first-principles model of Eq. 2.44 (i.e., $S \leq S_{TH}$ in the blue region and $S > S_{TH}$ in the white regions).

LMPC, the performance index S of Eq. 2.46 indicates the closeness of the convergence speed of closed-loop states between the LMPC using the data-driven model and the LMPC using the first-principles model.

By setting the threshold S_{TH} of the performance index to be 0.65, the region of initial conditions for which the performance of the ensemble of RNN models is close to that of the first-principles model (i.e., $S \leq S_{TH}$) covers the entire closed-loop stability region, while the corresponding region for the linear state-space model is characterized as the blue region in Fig. 2.15. It is shown in Fig. 2.15 that the closed-loop performance of the CSTR of Eq. 2.44 under the LMPC using the state-space model of Eq. 2.45 is undesired in the top and bottom of the closed-loop stability region due to poor approximation of nonlinearities in these regions. Therefore, based on the performance index of Eq. 2.46, the overall closed-loop performance of the ensemble of RNN models within the closed-loop stability region $\Omega_{\hat{\rho}}$ outperforms that of the state-space model in terms of the rate of convergence to the origin and the closeness to the closed-loop performance under the LMPC using the first-principles model.

2.4 Conclusions

This chapter presented a new class of model predictive controllers that utilize an ensemble of recurrent neural network models as the prediction model for nonlinear systems. Specifically, an RNN ensemble was first developed based on the dataset of extensive open-loop simulations within the operating region. Parallel computing was utilized to reduce the computation time of prediction by multiple RNN models. Then, the LMPC scheme that incorporate RNN models and Lyapunov-based stability constraints was formulated to stabilize a nonlinear process within the closed-loop stability region while optimizing process economic benefits simultaneously. The proposed LMPC using an RNN ensemble was applied to a nonlinear chemical process example to demonstrate its effectiveness.

Chapter 3

Machine-Learning-based Economic MPC

Since operational efficiency and increasing energy consumption are becoming crucially important issues in the chemical and petrochemical industry, a model-based feedback control strategy, economic model predictive control (EMPC), has been proposed as an efficient method to address process control problems integrated with dynamic economic optimization of the process, e.g., [14, 43, 57]. EMPC allows the chemical process to be operated in a time-varying fashion (off steady-state) to dynamically optimize process economic performance, and incorporates constraints that guarantee closed-loop stability and feasibility within an explicitly-defined estimate of the closed-loop stability region under an appropriate control law (e.g., a Lyapunov-based feedback control law).

In this chapter, we continue the discussion of the use of machine learning techniques in MPC, and develop the Lyapunov-based economic MPC (LEMPC) that incorporates an ensemble of RNN models to predict future states for the nonlinear system of Eq. 2.1. In the following subsections, a Lyapunov-based controller using the RNN model of Eq. 2.4 is first utilized to characterize the closed-loop stability region in which the origin of the nonlinear system of Eq. 2.1 can be rendered exponentially stable. Then, the formulation of the LEMPC using an ensemble of RNN models is given and Theorem 3.1 is established to demonstrate closed-loop stability for all initial conditions in the closed-loop stability region.

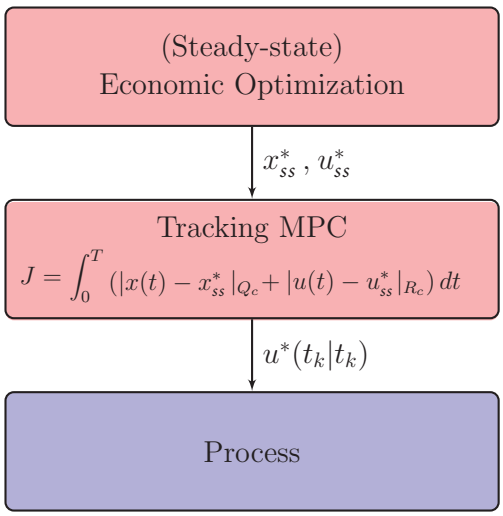


Figure 3.1: A two-layer paradigm for optimizing process economics within process control.

3.1 Economic Model Predictive Control

The economic success of the chemical and petrochemical industry relies on optimal process operation which has led to the emergence of an overall process control goal of incorporating process/system economic considerations into feedback control objectives. A traditional paradigm for optimizing process economics is to employ a two-layer control architecture as shown in Fig. 3.1, where in the upper layer, a real-time optimization (RTO) is solved to compute economically optimal steady-states that are sent to the lower layer, while in the lower layer, tracking MPC or traditional proportional-integral-derivative (PID) control is used to drive the process state to the optimal steady-state by computing optimal control actions u^*.

Another approach to addressing process control problems integrated with dynamic economic optimization of the process is to use economic model predictive control (EMPC). EMPC is a model-based feedback control technique that operates processes in a time-varying fashion (off steady-state) to dynamically optimize process economic performance, and incorporates constraints that guarantee closed-loop stability and feasibility within an explicitly-defined estimate of the closed-loop stability region under an appropriate control law (e.g., a Lyapunov-based feedback control law $\Phi(x)$). The EMPC that incorporates Lyapunov-based constraints in the design is termed Lyapunov-based economic MPC (LEMPC) and is represented by the following optimization

problem:

$$\max_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(\tau),u(\tau))\,d\tau \tag{3.1a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t),u(t),0) \tag{3.1b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{3.1c}$$

$$u(t) \in U,\ \forall\, t \in [t_k,t_{k+N}) \tag{3.1d}$$

$$V(\tilde{x}(t)) \leq \rho_e,\ \forall\, t \in [t_k,t_{k+N}),\ \text{if } x(t_k) \in \Omega_{\rho_e} \tag{3.1e}$$

$$\frac{\partial V(x(t_k))}{\partial x} f(x(t_k),u(t_k),0) \leq \frac{\partial V(x(t_k))}{\partial x} f(x(t_k),\Phi(x(t_k)),0),$$

$$\text{if } x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_e} \tag{3.1f}$$

where the notations follow those in Eq. 2.18 and Eq. 2.19. The LEMPC optimization problem maximizes the time-integral of the objective function $l_e(x,u)$ that represents the process economic performance over the prediction horizon subject to the constraints of Eqs. 3.1b-3.1f. Specifically, the constraints of Eqs. 3.1b-3.1d are the same as Eqs. 2.19b-2.19d for LMPC. The constraint of Eq. 3.1e (Mode 1 constraint) maintains the predicted states $\tilde{x}$ within Ω_{ρ_e} that is designed to ensure forward invariance of the closed-loop stability region Ω_ρ accounting for the sample-and-hold implementation of control actions and the impact of sufficiently small disturbances w, if the current state $x(t_k)$ at time $t = t_k$ is within Ω_{ρ_e}. However, if the current state leaves Ω_{ρ_e} due to disturbances, the constraint of Eq. 3.1f (Mode 2 constraint) is activated to drive the state towards the origin at least at the speed under the Lyapunov-based controller $\Phi(x(t_k))$ at $t = t_k$ such that it can enter Ω_{ρ_e} within finite sampling steps. An illustration of the closed-loop state trajectory under LEMPC is shown in Fig. 3.2.

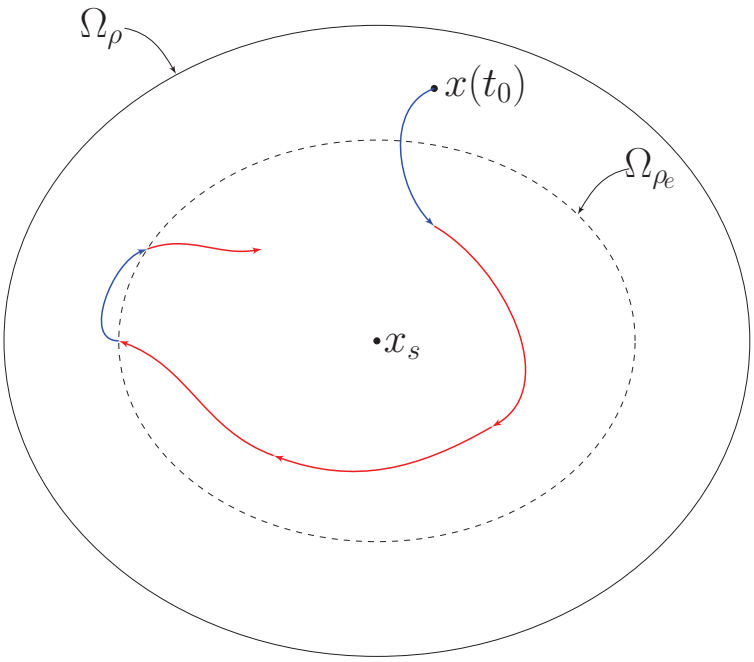


Figure 3.2: A state-space illustration of a closed-loop state trajectory under LEMPC, where the red and the blue trajectories are under Mode 1 and Mode 2 constraints, respectively.

3.2 Lyapunov-based EMPC using Ensemble RNN models

3.2.1 LEMPC Using an Ensemble of RNN Models

The Lyapunov-based economic MPC (LEMPC) design using an ensemble of RNN models is represented by the following optimization problem:

$$\mathscr{J} = \max_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(t), u(t)) dt \tag{3.2a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = \frac{1}{N_e} \sum_{j=1}^{N_e} F_{nn}^j(\tilde{x}(t), u(t)) \tag{3.2b}$$

$$u(t) \in U, \ \forall \ t \in [t_k, t_{k+N}) \tag{3.2c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{3.2d}$$

$$\hat{V}(\tilde{x}(t)) \leq \hat{\rho}_e, \ \forall \ t \in [t_k, t_{k+N}),$$

$$\text{if } x(t_k) \in \Omega_{\hat{\rho}_e} \tag{3.2e}$$

$$\dot{\hat{V}}(x(t_k), u) \leq \dot{\hat{V}}(x(t_k), \Phi_{nn}(x(t_k))),$$

$$\text{if } x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e} \tag{3.2f}$$

where $S(\Delta)$ is the set of piecewise constant functions with period Δ, $\tilde{x}$ is the predicted state trajectory, N_e is the number of RNN models used in predicting future states, and N is the number of sampling periods in the prediction horizon. We use $\dot{\hat{V}}(x,u)$ to represent the time-derivative of $\hat{V}$, i.e., $\dot{\hat{V}}(x,u) = \frac{\partial \hat{V}(x)}{\partial x}(F_{nn}(x,u))$. The optimization problem of Eq. 3.2 is solved by optimizing the time integral of the stage cost function $l_e(\tilde{x}(t),u(t))$ of Eq. 3.2a over the prediction horizon accounting for the constraints of Eqs. 3.2b-3.2f. The constraint of Eq. 3.2b is the ensemble of RNN models of Eq. 2.4 that is used as the prediction model. Eq. 3.2c defines the input constraints that are applied over the entire prediction horizon. Eq. 3.2d defines the initial condition $\tilde{x}(t_k)$ of Eq. 3.2b. The constraint of Eq. 3.2e maintains the closed-loop state predicted by Eq. 3.2b in $\Omega_{\hat{\rho}_e}$ over the prediction horizon if the state $x(t_k)$ is inside $\Omega_{\hat{\rho}_e}$. However, if $x(t_k)$ leaves $\Omega_{\hat{\rho}_e}$ but still remains in $\Omega_{\hat{\rho}}$, the contractive constraint of Eq. 3.2f drives the state towards the origin for the next sampling period such that the state will eventually enter $\Omega_{\hat{\rho}_e}$ within finite sampling periods. By introducing a conservative region $\Omega_{\hat{\rho}_e}$ with $\hat{\rho}_e < \hat{\rho}$ and the stability constraints of Eq. 3.2e and Eq. 3.2f, it will be proved in Theorem 3.1 that for any initial condition $x_0 \in \Omega_{\hat{\rho}}$, the closed-loop state of the nonlinear system of Eq. 2.1 is bounded in $\Omega_{\hat{\rho}}$ for all times. A schematic of $\Omega_{\hat{\rho}}$ and $\Omega_{\hat{\rho}_e}$ is given by Fig. 3.3, in which it is shown that the state trajectory starting inside $\Omega_{\hat{\rho}}$ is bounded in $\Omega_{\hat{\rho}}$ under LEMPC.

The optimal input trajectory is calculated over the entire prediction horizon $t \in [t_k, t_{k+N})$ and is denoted by $u^*(t)$. However, only the control action $u^*(t_k)$ for the first sampling period in $t \in [t_k, t_{k+N})$ is sent to the nonlinear system of Eq. 2.1 to be applied over the next sampling period. Before we demonstrate closed-loop stability for the nonlinear system of Eq. 2.1 under the LEMPC of Eq. 3.2, the next two propositions are first derived to provide useful tools for proving closed-loop stability in Theorem 3.1. Specifically, the following proposition provides an upper bound for the state error in the presence of sufficiently small modeling error and bounded disturbances.

Proposition 3.1. *Consider the solution $x(t)$ of the nonlinear system $\dot{x} = F(x,u,w)$ of Eq. 2.1 in the presence of bounded disturbances $|w(t)| \leq w_m$, and the solution $\hat{x}(t)$ of the RNN model $\dot{\hat{x}} = F_{nn}(\hat{x},u)$ of Eq. 2.4 with the same initial condition $x_0 = \hat{x}_0 \in \Omega_{\hat{\rho}}$. If $x(t)$, $\hat{x}(t) \in \Omega_{\hat{\rho}}$, and the modeling error*

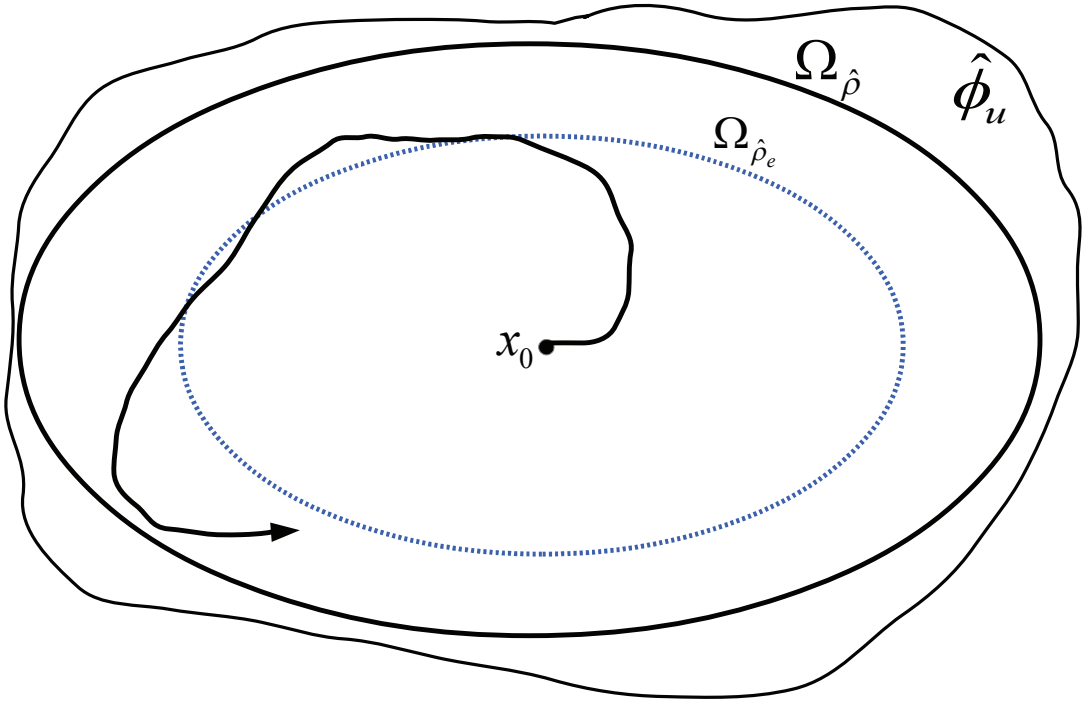


Figure 3.3: A schematic representing the set $\hat{\phi}_u$, the closed-loop stability region $\Omega_{\hat{\rho}}$, and the set $\Omega_{\hat{\rho}_e}$. Under the LEMPC of Eq. 3.2, the closed-loop state trajectory is bounded in $\Omega_{\hat{\rho}}$ for all times for any $x_0 \in \Omega_{\hat{\rho}}$.

is bounded (i.e., $|\nu(t)| = |\dot{x} - \dot{\hat{x}}| \leq \nu_m$*) for all times, then there exists a positive constant* κ *and a class* $\mathscr{K}$ *function* $f_w(\cdot)$ *such that the following inequalities hold* $\forall x, \hat{x} \in \Omega_{\hat{\rho}}$ *and* $w(t) \in W$*:*

$$|x(t) - \hat{x}(t)| \leq f_w(t) := \frac{L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1) \tag{3.3a}$$

$$\hat{V}(x) \leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{3.3b}$$

Proof. Following the proof in [6, 175], we define the state error vector by $e(t) = x(t) - \hat{x}(t)$. The time-derivative of $e(t)$ is obtained $\forall x, \hat{x} \in \Omega_{\hat{\rho}}$, $u \in U$ and $w(t) \in W$ as follows:

$$\begin{aligned} |\dot{e}| &= |F(x,u,w) - F_{nn}(\hat{x},u)| \\ &\leq |F(x,u,w) - F(\hat{x},u,0)| + |F(\hat{x},u,0) - F_{nn}(\hat{x},u)| \\ &\leq L_x|e(t)| + L_w w_m + \nu_m \end{aligned} \tag{3.4}$$

where the last inequality is derived from Eq. 2.21a and the fact that $|\nu| \leq \nu_m$. Since $x(t)$ and $\hat{x}(t)$ share the same initial condition, (i.e., $e(0) = 0$), the upper bound for $|e(t)|$ is derived for all $x(t), \hat{x}(t) \in \Omega_{\hat{\rho}}$ and $|w(t)| \leq w_m$ as follows:

$$|e(t)| = |x(t) - \hat{x}(t)| \leq \frac{L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1) \tag{3.5}$$

Subsequently, $\forall x, \hat{x} \in \Omega_\rho$, Eq. 3.3b is derived based on the Taylor series expansion of $\hat{V}(x)$ around $\hat{x}$ as follows:

$$\begin{aligned} \hat{V}(x) &\leq \hat{V}(\hat{x}) + \frac{\partial \hat{V}(\hat{x})}{\partial x}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \\ &\leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \end{aligned} \tag{3.6}$$

where κ is a positive real number, and the last inequality is derived using Eq. 2.20a and Eq. 2.20c.

□

The above proposition demonstrates that the error of state trajectories of the nonlinear system of Eq. 2.1 and the RNN model of Eq. 2.4, starting from the same initial condition, is bounded for

finite time. The next proposition is developed to demonstrate that if $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, the stabilizing controller $u = \Phi_{nn}(x) \in U$ implemented in a sample-and-hold fashion is able to maintain $\dot{\hat{V}}(x)$ negative such that the state will be driven towards the origin and ultimately enters a small neighborhood around the origin (i.e., Ω_{ρ_s}) in finite sampling steps.

Proposition 3.2. *Consider the system of Eq. 2.1 under the controller $u = \Phi_{nn}(\hat{x}) \in U$ that meets the conditions of Eq. 2.20 and is implemented in a sample-and-hold fashion, i.e., $u(t) = \Phi_{nn}(\hat{x}(t_k))$, $\forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$. Let $\varepsilon_w, \varepsilon_s > 0$, $\Delta > 0$ and $\hat{\rho} > \hat{\rho}_e > \rho_s > 0$ satisfy*

$$-\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{nn}M_{nn}\Delta \leq -\varepsilon_s \tag{3.7a}$$

$$-\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x^{'}M\Delta + L_w^{'}w_m \leq -\varepsilon_w \tag{3.7b}$$

$$\hat{\rho}_e > \max\{\hat{V}(\hat{x}(t_k+\Delta)) \mid \hat{x}(t_k) \in \Omega_{\rho_s}, u \in U, w \in W\} \tag{3.7c}$$

Then, for any $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, the following inequalities holds:

$$\hat{V}(\hat{x}(t)) \leq \hat{V}(\hat{x}(t_k)),\ \forall t \in [t_k, t_{k+1}) \tag{3.8a}$$

$$\hat{V}(x(t)) \leq \hat{V}(x(t_k)),\ \forall t \in [t_k, t_{k+1}) \tag{3.8b}$$

Proof. To show that the value of $\hat{V}$ decreases along the trajectory $\hat{x}(t)$ of the RNN model of Eq. 2.4 over $t \in [t_k, t_{k+1})$, we calculate the time-derivative of $\hat{V}(\hat{x})$ based on $\hat{x}(t)$ as follows:

$$\begin{aligned}
\dot{\hat{V}}(\hat{x}(t)) =& \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{nn}(\hat{x}(t), \Phi_{nn}(\hat{x}(t_k))) \\
=& \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{nn}(\hat{x}(t_k), \Phi_{nn}(\hat{x}(t_k))) \\
&+ \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{nn}(\hat{x}(t), \Phi_{nn}(\hat{x}(t_k))) \\
&- \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{nn}(\hat{x}(t_k), \Phi_{nn}(\hat{x}(t_k)))
\end{aligned} \tag{3.9}$$

Using Eq. 2.20a, 2.20b and the Lipschitz condition of Eq. 2.21, the following inequalities are

obtained:

$$\begin{aligned}\dot{\hat{V}}(\hat{x}(t)) \leq & -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{nn}(\hat{x}(t), \Phi_{nn}(\hat{x}(t_k))) \\ & - \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{nn}(\hat{x}(t_k), \Phi_{nn}(\hat{x}(t_k))) \\ \leq & -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{nn} M_{nn} \Delta \end{aligned} \tag{3.10}$$

Therefore, $\dot{\hat{V}}(\hat{x}(t)) \leq -\varepsilon_s$ holds $\forall \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k, t_{k+1})$ if Eq. 3.7a is satisfied. By integrating the above inequality, it follows that $\hat{V}(\hat{x}(t)) \leq \hat{V}(\hat{x}(t_k)) - \Delta\varepsilon_s$, $\forall \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, $t \in [t_k, t_{k+1})$ (i.e., Eq. 3.8a).

Next, to show that $\hat{V}(x(t)) \leq \hat{V}(x(t_k))$ holds for all $t \in [t_k, t_{k+1})$, the time-derivative of $\hat{V}(x)$ for the nonlinear system of Eq. 2.1 (i.e., $\dot{x} = F(x,u,w)$) in the presence of bounded disturbances (i.e., $|w(t)| \leq w_m$) is derived as follows:

$$\begin{aligned}\dot{\hat{V}}(x(t)) = & \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\ = & \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \\ & + \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\ & - \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \end{aligned} \tag{3.11}$$

Since Eq. 2.22 shows that $\frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \leq -\tilde{c}_3 |x(t_k)|^2$ holds for all $x \in \Omega_{\hat{\rho}}$, the following inequality is obtained for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, $t \in [t_k, t_{k+1})$ using Eq. 2.2a and the Lipschitz condition in Eq. 2.21:

$$\begin{aligned}\dot{\hat{V}}(x(t)) \leq & -\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x' |x(t) - x(t_k)| + L_w' |w| \\ \leq & -\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x' M \Delta + L_w' w_m \end{aligned} \tag{3.12}$$

Therefore, the following inequality is further derived for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k, t_{k+1})$ if the

condition of Eq. 3.7b is satisfied:

$$\dot{\hat{V}}(x(t)) \leq -\varepsilon_w \tag{3.13}$$

Similarly, this implies that $\hat{V}(x(t)) \leq \hat{V}(x(t_k)) - \Delta\varepsilon_w$, $\forall x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, $t \in [t_k, t_{k+1})$. Therefore, the state of the nonlinear system of Eq. 2.1 will enter Ω_{ρ_s} in finite sampling periods if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$. Additionally, if $x(t_k) \in \Omega_{\rho_s}$, in which Eq. 3.12 and Eq. 3.13 do not hold, Eq. 3.7c guarantees that the state will not leave $\Omega_{\hat{\rho}_e}$ in one sampling period for all $u \in U$ and $w \in W$. If the state $x(t_{k+1})$ leaves Ω_{ρ_s} but remains in $\Omega_{\hat{\rho}_e}$, at the next sampling period $t \in [t_{k+1}, t_{k+2})$, Eq. 3.13 is satisfied again and the state will be driven towards the origin. Therefore, the state of the nonlinear system of Eq. 2.1 is bounded in $\Omega_{\hat{\rho}}$ for all times. □

Based on Propositions 3.1 and 2.4, the following theorem is established to show recursive feasibility of the LEMPC optimization problem of Eq. 3.2, and the boundedness of the closed-loop state within $\Omega_{\hat{\rho}}$ under the sample-and-hold implementation of the LEMPC.

Theorem 3.1. *Consider the closed-loop system of Eq. 2.1 under the sample-and-hold implementation of the LEMPC of Eq. 3.2 with the stabilizing controller $\Phi_{nn}(x)$ that satisfies Eq. 2.20. Let $\Delta > 0$, $\varepsilon_w > 0$ and $\hat{\rho} > \hat{\rho}_e > 0$ satisfy Eq. 3.7 and the following inequality:*

$$\hat{\rho}_e \leq \hat{\rho} - \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}} f_w(\Delta) - \kappa(f_w(\Delta))^2 \tag{3.14}$$

If $x_0 \in \Omega_{\hat{\rho}}$ and the conditions of Proposition 3.1 and Proposition 3.2 are satisfied, then there always exists a feasible solution for the optimization problem of Eq. 3.2, and the closed-loop state $x(t)$ is bounded in the closed-loop stability region $\Omega_{\hat{\rho}}$, $\forall t \geq 0$.

Proof. The LEMPC of Eq. 3.2 predicts future states by averaging the results of an ensemble of RNN models in Eq. 3.2b. Since each RNN model is trained to satisfy the modeling error constraint and Eq. 2.21, it is readily shown that the averaged results of an ensemble of multiple RNN models also satisfy the above conditions, and thus, the results derived in Propositions 2.2-3.2 for a single

RNN model can be generalized to an RNN ensemble.

We first prove that the optimization problem of Eq. 3.2 can be solved with recursive feasibility for all $x \in \Omega_{\hat{\rho}}$. Specifically, if $x(t_k) \in \Omega_{\hat{\rho}_e}$, the control actions $\Phi_{nn}(x(t_{k+i}))$, $i = 0, 1, ..., N-1$ satisfy the input constraint of Eq. 3.2c and the Lyapunov-based constraint of Eq. 3.2e since it is shown in Eq. 3.8a that the states predicted by the RNN model of Eq. 3.2b remain inside $\Omega_{\hat{\rho}_e}$ under the controller $\Phi_{nn}(x)$. Additionally, if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$, the control action $u(t) = \Phi_{nn}(x(t_k)) \in U$, $t \in [t_k, t_{k+1})$ satisfies the input constraint of Eq. 3.2c and the Lyapunov-based constraint of Eq. 3.2f such that the state can be driven towards the origin during the next sampling period. Therefore, the stabilizing controller $u = \Phi_{nn}(x) \in U$ provides a feasible solution that satisfies all the constraints of the LEMPC optimization problem of Eq. 3.2 if $x(t) \in \Omega_{\hat{\rho}}$ for all times.

Next, we prove that for $x_0 \in \Omega_{\hat{\rho}}$, the state of the closed-loop system of Eq. 2.1 is bounded in $\Omega_{\hat{\rho}}$ for all times. Specifically, if $x(t_k) \in \Omega_{\hat{\rho}_e}$, the predicted states $\hat{x}(t)$ of the RNN model of Eq. 3.2b are maintained in $\Omega_{\hat{\rho}_e}$ under the constraint of Eq. 3.2e. According to Proposition 3.1, the actual state $x(t)$, $t \in [t_k, t_{k+1})$ of the nonlinear system of Eq. 2.1 is bounded by the following inequality:

$$\begin{aligned} \hat{V}(x) &\leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x-\hat{x}| + \kappa|x-\hat{x}|^2 \\ &\leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}} f_w(\Delta) + \kappa(f_w(\Delta))^2 \end{aligned} \tag{3.15}$$

Therefore, if $\Omega_{\hat{\rho}_e}$ is chosen as a level set of $\hat{V}$ that satisfies Eq. 3.14, $V(x)$ based on the actual state $x(t)$ is bounded in $\Omega_{\hat{\rho}}$ for all $t \in [t_k, t_{k+1})$. However, if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$, the constraint of Eq. 3.2f is activated such that the control action u decreases the value of $\hat{V}(\hat{x})$ based on the states predicted by the RNN model of Eq. 3.2b within the next sampling period. According to Eq. 3.8b in Proposition 3.2, the value of $\hat{V}$ also decreases along the state trajectory of the actual nonlinear system of Eq. 2.1 over $t \in [t_k, t_{k+1})$. Therefore, it is concluded that for any initial condition in $\Omega_{\hat{\rho}}$, the closed-loop state of the system of Eq. 2.1 is bounded in $\Omega_{\hat{\rho}}$ for all times under the LEMPC of Eq. 3.2. □

3.2.2 Application to a Chemical Process Example

We consider the same chemical reactor example as in Section 2.3.8 to illustrate the application of LEMPC using an ensemble of RNN models, under which the state of the closed-loop system is maintained within the stability region in state-space for all times. The CSTR is initially operated at the unstable steady-state $(C_{As},\ T_s) = (1.95\ kmol/m^3,\ 402\ K)$, and $(C_{A0_s}\ Q_s) = (4\ kmol/m^3,\ 0\ kJ/hr)$. In this example, the two manipulated inputs are the heat input rate, $\Delta Q = Q - Q_s$, and the inlet concentration of species A, $\Delta C_{A0} = C_{A0} - C_{A0_s}$, respectively. Additionally, the manipulated inputs are subject to the following constraints: $|\Delta Q| \leq 5 \times 10^5\ kJ/hr$ and $|\Delta C_{A0}| \leq 3.5\ kmol/m^3$. The states and the inputs of the system of Eq. 2.44 are represented by $x^T = [C_A - C_{As}\ T - T_s]$ and $u^T = [\Delta C_{A0}\ \Delta Q]$, respectively. Therefore, the unstable steady-state of the system of Eq. 2.44 is at the origin of the state-space, (i.e., $(x_s^*, u_s^*) = (0,0)$).

The control objective of the LEMPC of Eq. 3.2 is to maximize the production rate of B in the CSTR by manipulating the inlet concentration ΔC_{A0} and the heat input rate ΔQ, while maintaining the closed-loop state trajectories in the stability region $\Omega_{\hat{\rho}}$ for all times under LEMPC. The objective function of the LEMPC optimizes the production rate of B as follows:

$$l_e(\tilde{x}, u) = k_0 e^{-E/RT} C_A^2 \tag{3.16}$$

To simulate the dynamic model of Eq. 2.44 numerically, the explicit Euler method is utilized with an integration time step of $h_c = 10^{-4}\ hr$. The python module of the IPOPT software package [158], named PyIpopt, is utilized to solve the nonlinear optimization problem of the LEMPC of Eq. 3.2 with a sampling period $\Delta = 10^{-2}\ hr$. Additionally, a Message Passing Interface (MPI) for the Python programming language, named *MPI4Py* [40], is incorporated in the LEMPC optimization problem to execute multiple RNN predictions of Eq. 3.2b concurrently in independent computing processes.

Extensive open-loop simulations for the CSTR system of Eq. 2.44 are initially conducted to generate the dataset for RNN models. The control Lyapunov function $\hat{V}(x) = x^T P x$ is designed

with the following positive definite P matrix:

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{3.17}$$

Two hidden recurrent layers consisting of 96 and 64 recurrent units, respectively, are used in the RNN model. Additionally, a 10-fold cross validation are utilized to construct an ensemble of RNN models for the LEMPC of Eq. 3.2. The closed-loop stability region $\Omega_{\hat{\rho}}$ with $\hat{\rho} = 368$ and a subset $\Omega_{\hat{\rho}_e}$ with $\hat{\rho}_e = 320$ for the CSTR system are characterized based on the obtained RNN models and the stabilizing controller $u = \Phi_{nn}(x) \in U$. To demonstrate the effectiveness of the proposed LEMPC of Eq. 3.2, a linear state-space model (i.e., $\dot{x} = Ax + Bu$ with $A = 100 \times [-0.154 \;\; -0.003; \; 5.19 \;\; 0.138]$ and $B = [4.03 \;\; 0; \; 1.23 \;\; 0.004]$) is also identified following the system identification method in [74], and is added into the following performance comparison under the ensemble of RNN models and the first-principles model of Eq. 2.44.

The closed-loop simulation of the nominal CSTR system (i.e., $w(t) \equiv 0$) under the first-principles model of Eq. 2.44, the ensemble of RNN models, and the linear state-space model, respectively, are shown in Fig. 3.4. The following material constraint is utilized in the LEMPC of Eq. 3.2 to make the averaged reactant material available within the entire operating period t_p to be its steady-state value, C_{A0s} (i.e., the averaged reactant material in deviation form, u_1, is equal to 0).

$$\frac{1}{t_p}\int_0^{t_p} u_1(\tau) d\tau = 0 \; kmol/m^3 \tag{3.18}$$

In Fig. 3.4, it is shown that for the initial condition (0,0), the state trajectory under the ensemble of RNN models is close to that under the first-principles model, and both state trajectories are bounded in $\Omega_{\hat{\rho}_e}$ in the absence of disturbances. However, it is observed that the state trajectory under the linear state-space model ultimately leaves $\Omega_{\hat{\rho}}$ due to its large model mismatch. This implies that a more conservative set $\Omega_{\hat{\rho}_e}$ needs to be characterized for the state spaced model to guarantee the boundedness of state in the stability region $\Omega_{\hat{\rho}}$.

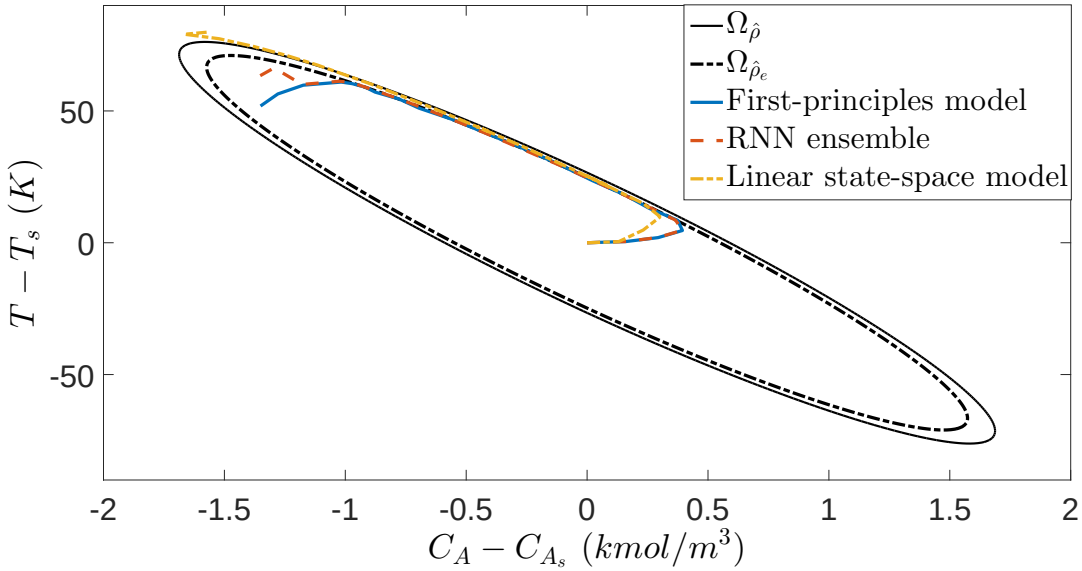


Figure 3.4: The state-space profiles for the closed-loop CSTR under the LEMPC using the following models: the first-principles model (blue trajectory), the RNN model ensemble (red trajectory) and the linear state-space model (yellow trajectory) for an initial condition (0, 0).

Additionally, Fig. 3.5 and Fig. 3.6 show the manipulated inputs profiles in deviation from the steady-state values. It is shown that the manipulated inputs are bounded within the input constraints for all times. Moreover, the material constraint of Eq. 3.18 is satisfied as it is shown in Fig. 3.5 that the LEMPC using the first-principles model and the RNN model ensemble both consume the maximum allowable ΔC_{A0} at the first few sampling steps, and thus have to lower the consumption of ΔC_{A0} near the end of operating period. The LEMPC using the linear state-space model also satisfies the material constraint, but shows persistent oscillation due to model mismatch. Additionally, in Fig 3.6, it is shown that the consumption of ΔQ under the RNN ensemble is close to that under the first-principles model (both correspond to left $y-axis$) since their closed-loop trajectories shown in Fig. 3.4 are similar. However, ΔQ shows large oscillation under the LEMPC using the linear state-space model (right $y-axis$) since the closed-loop trajectory in Fig. 3.4 leaves $\Omega_{\hat{\rho}_e}$, and thus, the contractive constraint of Eq. 3.2f is activated frequently.

To demonstrate that the closed-loop system under LEMPC achieves high process economics than the steady-state operation using the same amount reactant C_{A0}, we calculate the accumulated economic benefits $L_E = \int_0^{t_p} l_e(x,u)dt$ over the entire operating period $t_p = 0.2\ hr$, which are 2.04,

4.14, and 4.22 for the steady-state operation, the LEMPC under an ensemble of RNN models, and the LEMPC under the first-principles model, respectively. Therefore, it is demonstrated that both closed-loop stability and economic optimality are achieved under the proposed LEMPC using an ensemble of RNN models.

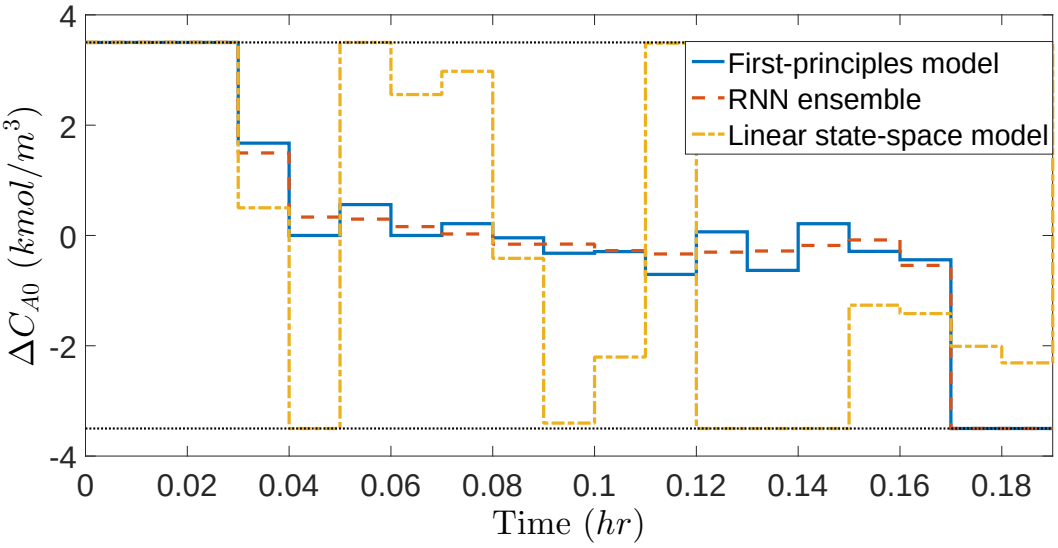


Figure 3.5: Manipulated input profiles ($u_1 = \Delta C_{A0}$) for the initial condition (0, 0) under the LEMPC using the following models: the first-principles model (blue trajectory), the RNN model ensemble (red trajectory) and the linear state-space model (yellow trajectory), where the dashed black horizontal lines represent the upper and lower bounds for ΔC_{A0}.

3.3 Conclusions

This chapter presented a new class of economic model predictive controllers that use an ensemble of recurrent neural network models as the prediction model for nonlinear systems. Using the RNN models developed in Chapter 2, the LEMPC that incorporates RNN models and Lyapunov-based stability constraints was formulated to stabilize a nonlinear process within the closed-loop stability region while optimizing process economic benefits simultaneously. The proposed LEMPC using an RNN ensemble was applied to a nonlinear chemical process example to demonstrate its economic optimality and closed-loop stability.

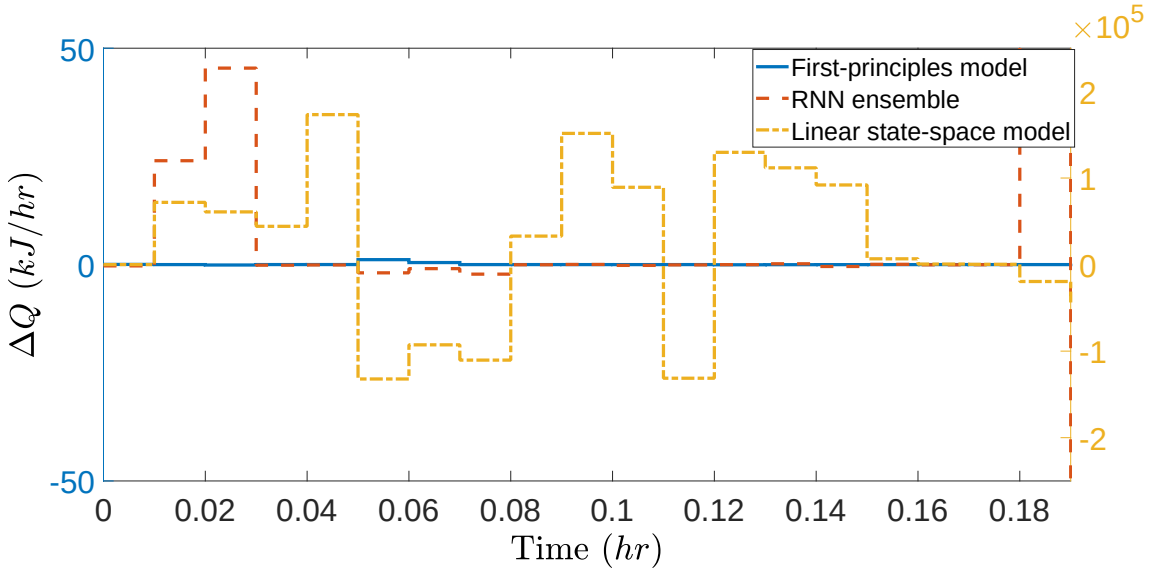


Figure 3.6: Manipulated input profiles ($u_2 = \Delta Q$) for the initial condition (0, 0) under the LEMPC using the following models: the first-principles model (blue trajectory corresponding to left $y-axis$), the RNN model ensemble (red trajectory corresponding to left $y-axis$) and the linear state-space model (yellow trajectory corresponding to right $y-axis$).

Chapter 4

Real-Time Machine-Learning-Based MPC

4.1 Introduction

Modeling large-scale, complex nonlinear processes has been a long-lasting challenge in process systems engineering. Model quality depends on many factors, including, but not limited to parameter estimation, model uncertainty, number of assumptions made in model development, dimensionality, model structure, and computational burden of solving the model in real-time operations ([47,48]). Machine learning techniques such as recurrent neural networks (RNN) have been successfully applied to solve regression/modeling problems based on datasets from industrial process operation or numerical simulations in process engineering, when a first-principles model is difficult to obtain. Although pre-trained machine learning models have demonstrated to be good replacements for first-principles models in model-based controllers, a potential problem for the real-time implementation of controllers in practice is model uncertainty, which includes intrinsic and exogenous uncertainty([93,151]).

Since in real life, processes models change in time due to varying process parameters from external (e.g., aging equipment, disturbance, and new implemented technology in the process) and internal factors (e.g., fouling in the equipment), the machine learning model that has been trained using the information from past normal operations may not be able to correctly predict process

states after disturbances appear.

Considering the need to update process models as time evolves, on-line learning of process models using most recent process data may provide a solution to deal with model uncertainty. It is noted that the event-triggered mechanism is able to reduce the frequency of on-line update and adjustment of process models and control actions ([144, 161]), and thus, improve applicability and efficiency of real-time control. For example, in [176], an event-based control was proposed to update the actuators only when a certain threshold is violated. In [161], an event-triggered mechanism was proposed to stabilize the system with control actions being updated when a violation of a stability event is triggered. Additionally, the event-triggered concept has also been adopted in neural network-based control to reduce the network source utilization [84, 134].

Motivated by the above, in this chapter, we propose real-time machine learning-based MPC and EMPC schemes that trigger an on-line learning of RNN models when a threshold is violated due to unknown disturbances. Specifically, an ensemble of RNN models is initially obtained for the Lyapunov-based MPC and EMPC to stabilize the system under normal operation (i.e., without disturbances). In the presence of time-varying disturbances, an event-triggered mechanism based on the decreasing rate of Lyapunov function and an error-triggered mechanism based on the prediction errors are developed to update RNN models during the operation using the most recent process data. Closed-loop stability analysis is provided for both LMPC and LEMPC with on-line RNN model update.

4.1.1 Notation

The notation $|\cdot|$ is used to denote the Euclidean norm of a vector. x^T denotes the transpose of x. The notation $L_f V(x)$ denotes the standard Lie derivative $L_f V(x) := \frac{\partial V(x)}{\partial x} f(x)$. Set subtraction is denoted by "\", i.e., $A \backslash B := \{x \in \mathbf{R}^n \mid x \in A, x \notin B\}$. $\mathbb{Z}^+$ denotes the set of positive integers. $\emptyset$ signifies the null set. The function $f(\cdot)$ is of class $\mathscr{C}^1$ if it is continuously differentiable in its domain. A continuous function $\alpha : [0, a) \to [0, \infty)$ is said to belong to class $\mathscr{K}$ if it is strictly increasing and is zero only when evaluated at zero.

4.1.2 Class of Systems

The class of continuous-time nonlinear systems considered is described by the following system of first-order nonlinear ordinary differential equations:

$$\dot{x} = F(x,u,w) := f(x) + g(x)u + h(x)w,\ x(t_0) = x_0 \tag{4.1}$$

where $x \in \mathbf{R}^n$ is the state vector, $u \in \mathbf{R}^m$ is the manipulated input vector, and $w \in W$ is the disturbance vector with $W := \{w \in \mathbf{R}^q \mid |w| \leq w_m,\ w_m \geq 0\}$. The control actions are constrained by $u \in U := \{u_i^{\min} \leq u_i \leq u_i^{\max}, i = 1,...,m\} \subset \mathbf{R}^m$. $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are sufficiently smooth vector and matrix functions of dimensions $n \times 1$, $n \times m$, and $n \times q$, respectively. Throughout this chapter, we assume that the initial time t_0 is zero ($t_0 = 0$), and $f(0) = 0$ such that the origin is a steady-state of the nominal (i.e., $w(t) \equiv 0$) system of Eq. 4.1 (i.e., $(x_s^*, u_s^*) = (0,0)$, where x_s^* and u_s^* represent the steady-state state and input vectors, respectively).

To guarantee that the closed-loop system is stabilizable, a stabilizing control law $u = \Phi(x) \in U$ that renders the origin of the nominal system of Eq. 4.1 (i.e., $w(t) \equiv 0$) exponentially stable is assumed to exist. Following converse theorems, there exists a $\mathscr{C}^1$ Control Lyapunov function $V(x)$ such that the following inequalities hold for all x in an open neighborhood D around the origin:

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2, \tag{4.2a}$$

$$\frac{\partial V(x)}{\partial x} F(x, \Phi(x), 0) \leq -c_3|x|^2, \tag{4.2b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq c_4|x| \tag{4.2c}$$

where c_1, c_2, c_3 and c_4 are positive constants. $F(x,u,w)$ represents the nonlinear system of Eq. 4.1. The universal Sontag control law ([83]) is a candidate controller for $u = \Phi(x)$.

We first characterize a region where the time-derivative of V is rendered negative under the

controller $u = \Phi(x) \in U$ as follows:

$$\phi_u = \{x \in \mathbf{R}^n \mid \dot{V}(x) = L_f V + L_g V u < -k|x|^2, u = \Phi(x) \in U\} \cup \{0\} \tag{4.3}$$

where k is a positive real number. Then a level set of the Lyapunov function inside ϕ_u is used as the closed-loop stability region Ω_ρ for the nonlinear system of Eq. 4.1 as follows: $\Omega_\rho := \{x \in \phi_u \mid V(x) \leq \rho\}$, where $\rho > 0$ and $\Omega_\rho \subset \phi_u$. From the Lipschitz property of $F(x,u,w)$ and the bounds on u and w, it follows that there exist positive constants M, $L_x, L_w, L_x^{'}, L_w^{'}$ such that the following inequalities hold for all $x, x' \in D, u \in U$, and $w \in W$:

$$|F(x,u,w)| \leq M \tag{4.4a}$$

$$|F(x,u,w) - F(x',u,0)| \leq L_x|x-x'| + L_w|w| \tag{4.4b}$$

$$\left|\frac{\partial V(x)}{\partial x}F(x,u,w) - \frac{\partial V(x')}{\partial x}F(x',u,0)\right| \leq L_x^{'}|x-x'| + L_w^{'}|w| \tag{4.4c}$$

4.1.3 Preliminary Results of RNN-based MPC

We first present some preliminary results of Lyapunov-based MPC (LMPC) and Lyapunov-based economic MPC (LEMPC) using RNN models. The recurrent neural network model is developed with the following form:

$$\dot{\hat{x}} = F_{nn}(\hat{x}, u) := A\hat{x} + \Theta^T y \tag{4.5}$$

where the notations follow those in Eq. 2.4, and the training process follows the same learning algorithm as in Chapter 2.

In this chapter, the RNN model of Eq. 4.5 is updated to capture nonlinear dynamics of the nonlinear system of Eq. 4.1 subject to time-varying bounded disturbances (i.e., $|w(t)| \leq w_m$). $F_{nn}^i(x,u)$ is used to denote the ith RNN model ($i = 1, 2, ..., N_T$) that is updated using the real-time data of closed-loop state trajectories and control actions, where N_T is the total number of RNN models obtained. We assume that a set of stabilizing feedback controllers $u = \Phi_{nn}^i(x) \in U$ that can

render the origin of the RNN models $F_{nn}^i(x,u)$, $i = 1,2,...,N_T$ of Eq. 4.5 exponentially stable in an open neighborhood $\hat{D}$ around the origin exists. Therefore, there exists a $\mathscr{C}^1$ Control Lyapunov function $\hat{V}(x)$ such that the following inequalities hold for all x in $\hat{D}$:

$$\hat{c}_1^i|x|^2 \leq \hat{V}(x) \leq \hat{c}_2^i|x|^2, \tag{4.6a}$$

$$\frac{\partial \hat{V}(x)}{\partial x} F_{nn}^i(x, \Phi_{nn}^i(x)) \leq -\hat{c}_3^i|x|^2, \tag{4.6b}$$

$$\left|\frac{\partial \hat{V}(x)}{\partial x}\right| \leq \hat{c}_4^i|x| \tag{4.6c}$$

where $\hat{c}_1^i$, $\hat{c}_2^i$, $\hat{c}_3^i$, $\hat{c}_4^i$ are positive constants, $i = 1,2,...,N_T$. For the sake of simplicity, we will use symbols without the superscript of i for all the RNN models and controllers that satisfy Eq. 4.6 in the following texts. Similar to the characterization method of the closed-loop stability region Ω_ρ for the nonlinear system of Eq. 4.1, we first characterize a region $\hat{\phi}_u = \{x \in \mathbf{R}^n \mid \dot{\hat{V}}(x) < -\hat{c}_3|x|^2, u = \Phi_{nn}(x) \in U\} \cup \{0\}$, from which the origin of the RNN model of Eq. 4.5 can be rendered exponentially stable under the controller $u = \Phi_{nn}(x) \in U$.

The closed-loop stability region for the RNN model of Eq. 4.5 is defined as a level set of Lyapunov function inside $\hat{\phi}_u$: $\Omega_{\hat{\rho}} := \{x \in \hat{\phi}_u \mid \hat{V}(x) \leq \hat{\rho}\}$, where $\hat{\rho} > 0$. It is noted that $\Omega_{\hat{\rho}} \subseteq \Omega_\rho$ since the dataset for developing the RNN model of Eq. 4.5 is generated from open-loop simulations for $x \in \Omega_\rho$ and $u \in U$. Additionally, there exist positive constants M_{nn} and L_{nn} such that the following inequalities hold for all $x, x' \in \Omega_{\hat{\rho}}$ and $u \in U$:

$$|F_{nn}(x,u)| \leq M_{nn} \tag{4.7a}$$

$$\left|\frac{\partial \hat{V}(x)}{\partial x} F_{nn}(x,u) - \frac{\partial \hat{V}(x')}{\partial x} F_{nn}(x',u)\right| \leq L_{nn}|x - x'| \tag{4.7b}$$

Consider that there exists a bounded modeling error between the nominal system of Eq. 4.1 and the RNN model of Eq. 4.5 (i.e., $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \nu_m$, $\nu_m > 0$), the following proposition demonstrates that the feedback controller $u = \Phi_{nn}(x) \in U$ is able to stabilize the

nominal system of Eq. 4.1 if the modeling error is sufficiently small.

Proposition 4.1. *Under the assumption that the origin of the closed-loop RNN system of Eq. 4.5 is rendered exponentially stable under the controller $u = \Phi_{nn}(x) \in U$ for all $x \in \Omega_{\hat{\rho}}$, if there exists a positive real number $\gamma < \hat{c}_3/\hat{c}_4$ that constrains the modeling error $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma|x| \leq \nu_m$ for all $x \in \Omega_{\hat{\rho}}$ and $u \in U$, then the origin of the nominal closed-loop system of Eq. 4.1 under $u = \Phi_{nn}(x) \in U$ is also exponentially stable for all $x \in \Omega_{\hat{\rho}}$.*

The formulation of the LMPC using an ensemble of RNN models is given as follows:

$$\mathscr{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), u(t)) dt \tag{4.8a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = \frac{1}{N_e} \sum_{j=1}^{N_e} F_{nn,j}(\tilde{x}(t), u(t)) \tag{4.8b}$$

$$u(t) \in U, \ \forall \ t \in [t_k, t_{k+N}) \tag{4.8c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{4.8d}$$

$$\dot{\hat{V}}(x(t_k), u) \leq \dot{\hat{V}}(x(t_k), \Phi_{nn}(x(t_k))), \text{ if } x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{nn}} \tag{4.8e}$$

$$\hat{V}(\tilde{x}(t)) \leq \rho_{nn}, \ \forall \ t \in [t_k, t_{k+N}), \text{ if } x(t_k) \in \Omega_{\rho_{nn}} \tag{4.8f}$$

where the notation follows that in Eq. 2.41.

The Lyapunov-based economic MPC (LEMPC) using an ensemble of RNN models is developed to dynamically optimize process economic benefits while maintaining closed-loop state in the

stability region for all times. The LEMPC is represented by the following optimization problem:

$$\mathscr{J} = \max_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(t), u(t)) dt \tag{4.9a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = \frac{1}{N_e} \sum_{j=1}^{N_e} F_{nn,j}(\tilde{x}(t), u(t)) \tag{4.9b}$$

$$u(t) \in U, \ \forall \ t \in [t_k, t_{k+N}) \tag{4.9c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{4.9d}$$

$$\hat{V}(\tilde{x}(t)) \leq \hat{\rho}_e, \ \forall \ t \in [t_k, t_{k+N}), \ \text{if } x(t_k) \in \Omega_{\hat{\rho}_e} \tag{4.9e}$$

$$\dot{\hat{V}}(x(t_k), u) \leq \dot{\hat{V}}(x(t_k), \Phi_{nn}(x(t_k))), \ \text{if } x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e} \tag{4.9f}$$

where the notation follows that in Eq. 3.2.

4.2 Event-triggered On-line Learning of RNNs

In this section, the LMPC of Eq. 4.8 and the LEMPC of Eq. 4.9 are applied to the nonlinear system of Eq. 4.1 subject to bounded disturbances (i.e.,$|w(t)| \leq w_m$). Unlike the stability analysis performed for sufficiently small bounded disturbances in [171, 181], in this chapter, we consider the case in which disturbances cannot be fully eliminated by the sample-and-hold implementation of LMPC and therefore, may render the closed-loop system unstable. To mitigate the impact of disturbances, RNN models are updated via on-line learning to capture the nonlinear dynamics of the system of Eq. 4.1 accounting for disturbances $w(t)$. In the following subsections, the triggering mechanisms for updating RNN models are introduced.

4.2.1 Event-triggering Mechanism

In [56], event-triggered and self-triggered control systems were introduced to derive closed-loop stability for the system under the sample-and-hold implementation of a controller. Specifically, the event-triggered control system triggers an update of control actions if a triggering condition based

on state measurements is violated, while in self-triggered control system, the next update time can be obtained via predictions. In our work, an event-triggered on-line RNN learning is incorporated in the LMPC of Eq. 4.8 and the LEMPC of Eq. 4.9 to improve RNN prediction accuracy using previously received data of closed-loop states in the presence of bounded disturbances. The following theorem is established to demonstrate that if the on-line update of RNN is triggered by the violation of Eq. 4.10, the minimal inter-event time $T_k = r_{k+1} - r_k$ is bounded from below, where r_k represents the kth violation of Eq. 4.10, $k \in \mathbb{Z}^+$.

Theorem 4.1. *Consider the nonlinear system $\dot{x} = F(x,u,w)$ of Eq. 4.1 in the presence of bounded disturbances $|w(t)| \leq w_m$, and the RNN model $\dot{\hat{x}} = F_{nn}(\hat{x},u)$ of Eq. 4.5 that has been updated at $t = t_k = r_k$ to approximate dynamic behavior of the system of Eq. 4.1 before $t = t_k$ with a sufficiently small modeling error $|\nu| \leq \gamma|x|$, $\gamma < \hat{c}_3/\hat{c}_4$. If the stabilizing controller $u = \Phi_{nn}(x) \in U$ is implemented in a sample-and-hold fashion (i.e., $u(t) = \Phi_{nn}(\hat{x}(t_k))$, $\forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$ and Δ is the sampling period), and the $k+1$th update of RNN model is triggered at $t = r_{k+1}$ by the violation of the following inequality for all $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$:*

$$V(x(t)) \leq V(x(t_k)) - \varepsilon_w(t - t_k),\ t \in [t_k, t_{k+1}) \tag{4.10}$$

where $\varepsilon_w > 0$ and ρ_s satisfy Eq. 2.30 and Eq. 2.31 in Theorem 2.1, then there exists a positive constant τ^* such that the minimal inter-event time $T_k = r_{k+1} - r_k \geq \tau^*$.

Proof. Since the controller $u = \Phi_{nn}(x) \in U$ is implemented in sample-and-hold fashion, given $x(t_k) = \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, we first derive the time-derivative of $\hat{V}(x)$ for the nonlinear system of Eq. 4.1 (i.e., $\dot{x} = F(x,u,w)$) in the presence of bounded disturbances (i.e., $|w| \leq w_m$) over $t \in$

$[t_k, t_{k+1})$ as follows:

$$\begin{aligned}\dot{\hat{V}}(x(t)) =& \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w(t)) \\ =& \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), w(t_k)) \\ &+ \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w(t)) \\ &- \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), w(t_k))\end{aligned} \tag{4.11}$$

The first term $\frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), w(t_k))$ in the above equation can be further expanded as follows:

$$\begin{aligned}&\frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), w(t_k)) \\ &= \frac{\partial \hat{V}(x(t_k))}{\partial x} (F_{nn}(x, \Phi_{nn}(x(t_k))) \\ &\quad + F(x, \Phi_{nn}(x(t_k)), w(t_k)) - F_{nn}(x, \Phi_{nn}(x(t_k)))) \\ &\leq -\hat{c}_3 |x(t_k)|^2 + \hat{c}_4 |x(t_k)| (F(x(t_k), \Phi_{nn}(x(t_k)), w(t_k)) \\ &\quad - F_{nn}(x(t_k), \Phi_{nn}(x(t_k)))) \\ &\leq -\hat{c}_3 |x(t_k)|^2 + \hat{c}_4 \gamma |x(t_k)|^2 \\ &\leq -\tilde{c}_3 |x(t_k)|^2\end{aligned} \tag{4.12}$$

where $\tilde{c}_3 = -\hat{c}_3 + \hat{c}_4 \gamma > 0$ is a positive real number that has been defined in Theorem 2.1. Specifically, the inequalities in Eq. 4.12 are derived from the fact that $\frac{\partial \hat{V}(x(t))}{\partial x} F_{nn}(x(t), \Phi_{nn}(x)) \leq -\hat{c}_3 |x(t)|^2$ holds for all $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, and the RNN model $\dot{\hat{x}} = F_{nn}(\hat{x}, u)$ is well-trained at $t = t_k$ such that the modeling error $|\nu| = |F(x, u, w) - F_{nn}(\hat{x}, u)|$, $\forall t \in [0, t_k]$ is constrained by $|\nu| \leq \gamma |x|$. Based on Eq. 4.12 and Eq. 4.4, the time-derivative of $\hat{V}$ in Eq. 4.11 can be simplified as follows:

$$\begin{aligned}\dot{\hat{V}}(x(t)) \leq& -\tilde{c}_3 |x(t_k)|^2 + \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w(t)) \\ &- \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), w(t_k)) \\ \leq& -\tilde{c}_3 |x(t_k)|^2 + L_x' |x(t) - x(t_k)| + L_w' |w(t) - w(t_k)|\end{aligned} \tag{4.13}$$

Let $p(t) = \hat{V}(x(t))$ and $q(t) = \tilde{V}(x(t)) = \hat{V}(x(t_k)) - \varepsilon_w(t - t_k)$. It is readily shown that $p(t)$ and $q(t)$ are $\mathscr{C}^1$ functions and $p(t_k) = q(t_k) = \hat{V}(x(t_k))$ holds. It follows that $\dot{q}(t_k) = \dot{\tilde{V}}(x(t_k)) = -\varepsilon_w$. Additionally, using Eq. 4.2 and Eq. 4.13, $\dot{p}(t_k)$ is bounded by the following inequality:

$$\begin{aligned} \dot{p}(t_k) = \dot{\hat{V}}(x(t_k)) &\leq -\tilde{c}_3|x(t_k)|^2 + L_x^{'}|x(t_k) - x(t_k)| + L_w^{'}|w(t_k) - w(t_k)| \\ &\leq -\frac{\tilde{c}_3}{\hat{c}_2}\hat{V}(x(t_k)) \end{aligned} \tag{4.14}$$

Therefore, it is derived that $\dot{p}(t_k) < \dot{q}(t_k)$ for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ since ε_w is chosen to satisfy Eq. 2.30 (i.e., $-\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x^{'}M\Delta \leq -\varepsilon_w$). Following the Lemma in [161], it is shown that the minimal inter-event time T_k satisfies $T_k \geq \tau^*$, where τ^* is the smallest positive solution to the equation $p(t) = q(t)$, due to the continuity properties of $p, q, \dot{p}, \dot{q}$. This completes the proof of Theorem 4.1. □

Remark 4.1. *Theorem 4.1 demonstrates that the existence of a nonzero minimal inter-event time T_k is guaranteed for the nonlinear system of Eq. 4.1 subject to the triggering condition of Eq. 4.10. This implies that the above sample-and-hold implementation of the controller $u = \Phi_{nn}(x) \in U$ with the triggering condition of Eq. 4.10 can be applied in practice in which the update of RNN models cannot be triggered in a continuous-time manner.*

Remark 4.2. *Since the upper bound of the evolution of $V(x)$ given in Eq. 4.10 guarantees the decrease of $V(x)$ over time, the closed-loop state can be ultimately driven into a small neighborhood around the origin (i.e., Ω_{ρ_s}) under the controller $u = \Phi_{nn}(x) \in U$ provided that the RNN models of Eq. 4.5 and control actions are updated every time the condition of Eq. 4.10 is violated (i.e., at $t = r_k$, $k = 1, 2, ...$). However, considering the fixed sampling period Δ in the sample-and-hold implementation of the LMPC of Eq 4.8 and the LEMPC of Eq 4.9, control actions based on the updated RNN models will not be calculated immediately after the violation of Eq. 4.10 since the control actions remain the same during the current sampling period. For example, if the $(k+1)$th RNN update is triggered at $t = r_{k+1}$, where $r_{k+1} \in (t_k, t_{k+1})$, the control actions are calculated based on the new RNN models at the next sampling time $t = t_{k+1}$ instead of $t = r_{k+1}$. Due to the asynchronization between updating RNN models and re-calculating control actions, Eq. 4.10*

may not hold for all times, and thus, the closed-loop state is no longer guaranteed to move towards the origin within each sampling period. To address the above issue, an additional constraint is proposed for the sampling period in the following subsection to ensure that the closed-loop state can still be driven to a neighborhood around the origin under asynchronous updates of RNN models and control actions.

4.2.2 Stability Analysis of Event-triggered Feedback Systems

Since model uncertainty (i.e., bounded disturbances $|w(t)| \leq w_m$) is introduced into the nonlinear system of Eq. 4.1 under the sample-and-hold implementation of the controller $u = \Phi_{nn}(x) \in U$ that incorporates the event-triggered mechanism of Eq. 4.10, closed-loop stability derived for the nominal system of Eq. 4.1 does not hold for all x in $\Omega_{\hat{\rho}}$. In this section, we show that the controller $u = \Phi_{nn}(x) \in U$ can maintain the state inside the stability region $\Omega_{\hat{\rho}}$ for all times and ultimately drive the state into a region around the origin for the closed-loop system of Eq. 4.1 subject to bounded disturbances.

The following proposition is developed to demonstrate that if the RNN model update is triggered within a certain sampling period, yet the control actions remain unchanged till the end of this sampling period, closed-loop stability is still guaranteed in the sense that the closed-loop state moves towards the origin within one sampling period for all $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$, where $\rho_w \geq \max_{x \in \Omega_{\hat{\rho}}} \{\hat{V}(x) \mid \dot{\hat{V}}(x) \geq -\hat{c}_3|x|^2 - 2L_w^{'} w_m, u = \Phi(x) \in U\}$. Additionally, ρ_w is designed such that if the current state is inside Ω_{ρ_w}, it will not leave $\Omega_{\hat{\rho}}$ within one sampling period.

Proposition 4.2. *Consider the system of Eq. 4.1 with bounded disturbances (i.e., $|w(t)| \leq w_m$) under the sample-and-hold implementation of the controller $u = \Phi_{nn}(x) \in U$. Let $\hat{\rho} > \rho_w > 0$ and Δ satisfy Eq. 2.30 and the following inequality:*

$$\Delta < \frac{2(\frac{\tilde{c}_3}{\hat{c}_2}\rho_w - 2L_w^{'} w_m)}{L_x^{'} M} \tag{4.15}$$

Then, for any $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$, it holds that

$$\hat{V}(x(t)) < \hat{V}(x(t_k)),\ \forall t \in (t_k, t_{k+1}] \tag{4.16}$$

Proof. Assuming $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$, we prove that within one sampling period, the value of $\hat{V}(x(t))$ does not exceed that of $\hat{V}(x(t_k))$ for all $t \in [t_k, t_{k+1}]$ in the case that the RNN model updated at $t = r_k < t_k$ does not account for current disturbances $w(t)$ at all. Based on Eq. 4.11, the time-derivative of $\hat{V}(x)$ in the presence of disturbances is derived as follows:

$$\begin{aligned}
\dot{\hat{V}}(x(t)) =& \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w(t)) \\
=& \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), w(r_k)) \\
&+ \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w(t)) \\
&- \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), w(r_k))
\end{aligned} \tag{4.17}$$

Using the similar expansion that has been performed in Eq. 4.12, we derive the following equation:

$$\begin{aligned}
&\frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), w(r_k)) \\
=& \frac{\partial \hat{V}(x(t_k))}{\partial x} (F_{nn}(x, \Phi_{nn}(x(t_k))) \\
&+ F(x, \Phi_{nn}(x(t_k)), w(r_k)) - F_{nn}(x, \Phi_{nn}(x(t_k))))
\end{aligned} \tag{4.18}$$

Since the RNN model obtained at $t = r_k$ guarantees that the modeling error between the kth RNN model and the uncertain nonlinear system of Eq. 4.1 subject to disturbances $w(t)$, $\forall t \in [r_{k-1}, r_k]$, is sufficiently small (i.e., $|F(x, \Phi_{nn}(x), w(r_k)) - F_{nn}(x, \Phi_{nn}(x))| \leq \gamma |x|$), the following inequalities

can be obtained using Eq. 4.4:

$$\begin{aligned}\dot{\hat{V}}(x(t)) \leq & -\hat{c}_3|x(t_k)|^2+\hat{c}_4\gamma|x(t_k)|^2+L_x^{'}|x(t)-x(t_k)| \\ & +L_w^{'}|w(t)-w(r_k)| \\ \leq & -\tilde{c}_3|x(t_k)|^2+L_x^{'}|x(t)-x(t_k)|+L_w^{'}|w(t)-w(r_k)|\end{aligned} \tag{4.19}$$

□

From the above inequality, it is noted that the disturbance term $|w(t)-w(r_k)|$ could be nonzero for all $t \in [t_k, t_{k+1}]$ because the last updated RNN model (i.e., the kth RNN model obtained at $t = r_k < t_k$) does not account for time-varying disturbances over $t \in (r_k, t_k]$. Therefore, we show that Eq. 4.16 holds for all $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$ under the worst-case scenario that $|w(t)-w(r_k)| = 2w_m$, $\forall t \in [t_k, t_{k+1}]$. Specifically, based on Eq. 4.17, Eq. 4.18 and the fact that $\frac{\partial \hat{V}(x(t_k))}{\partial x} F_{nn}(x, \Phi_{nn}(x(t_k))) < -\hat{c}_3|x(t_k)|^2 - 2L_w^{'}w_m$ for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$, it is obtained that

$$\begin{aligned}\dot{\hat{V}}(x(t_k)) \leq & -\hat{c}_3|x(t_k)|^2-2L_w^{'}w_m+\hat{c}_4\gamma|x(t_k)|^2 \\ & +L_x^{'}|x(t_k)-x(t_k)|+L_w^{'}|w(t_k)-w(r_k)| \\ \leq & -\frac{\tilde{c}_3}{\hat{c}_2}\rho_w\end{aligned} \tag{4.20}$$

It follows that $x(t)$ initially moves towards the origin during $t \in [t_k, t_{k+1}]$ due to $\dot{\hat{V}}(x(t_k)) < 0$ at $t = t_k$. Next, we show that $\hat{V}(x(t)) < \hat{V}(x(t_k))$ holds for all $t \in (t_k, t_{k+1}]$ provided that the sampling period Δ is sufficiently small. From Eq. 4.4a and Eq. 4.19, it is obtained that $\dot{\hat{V}}(x(t)) \leq -\frac{\tilde{c}_3}{\hat{c}_2}\hat{V}(x(t_k)) + L_x^{'}M|t-t_k| + 2L_w^{'}w_m$, $\forall t \in [t_k, t_{k+1}]$. Thus, the evolution of $\hat{V}(x(t))$, $t \in [t_k, t_{k+1}]$ is calculated as follows by letting $\tau = t - t_k$:

$$\hat{V}(x(t)) \leq V(x(t_k)) + (2L_w^{'}w_m - \frac{\tilde{c}_3}{\hat{c}_2}\rho_w)\tau + \frac{L_x^{'}M}{2}\tau^2 \tag{4.21}$$

Therefore, if the sampling period satisfies Eq. 4.15, it is guaranteed that $\hat{V}(x(t)) < \hat{V}(x(t_k))$ for all $t \in (t_k, t_{k+1}]$, where $t_{k+1} := t_k + \Delta$. This implies that for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$, the state is bounded in

$\Omega_{\hat{\rho}}$ for all times, and can be ultimately driven into Ω_{ρ_w} under $u = \Phi_{nn}(x) \in U$.

Remark 4.3. *Although the controller $u = \Phi_{nn}(x) \in U$ is able to drive the state towards the origin for all $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$, the rate of convergence could be slow due to the large model mismatch if the RNN models are not updated following the event-triggering mechanism of Eq. 4.10. Therefore, to accelerate convergence, it is necessary for the RNN models to be updated on-line to improve approximation performance. For example, the on-line update of the $k+1$th RNN model is triggered at $t = r_{k+1}$ to capture the dynamics of the nonlinear system of Eq.4.1 accounting for time-varying disturbances since the last update invocation (i.e., $t \in [r_k, r_{k+1}]$). As a result, the new RNN models work compatibly with the controller to stabilize the nonlinear system of Eq. 4.1 until the model mismatch increases to an undesired level and eventually leads to the next violation of Eq. 4.10.*

Remark 4.4. *Suppose that an on-line update of RNN models is triggered at some point within one sampling period (e.g., $r_k \in (t_k, t_{k+1})$). Since the control actions remain unchanged till the next sampling step t_{k+1} due to the sample-and-hold implementation of the controller, Proposition 4.2 demonstrates that for any $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$, the state $x(t)$, $\forall (t_k, t_{k+1}]$ can still move towards a smaller level set of $\hat{V}(x)$ if the sampling period Δ satisfies Eq. 4.15. The above stability property facilitates and eases the incorporation of the event-triggered update of RNN models into the LMPC of Eq. 4.8 and the LEMPC of Eq 4.9 where a fixed sampling period Δ is used.*

4.2.3 Error-triggered On-line RNN Update

The above sections have demonstrated that the closed-loop state of the system of Eq. 4.1 subject to bounded disturbances can be driven into Ω_{ρ_w} under $u = \Phi_{nn}(x) \in U$ with on-line update of RNN models. Since $\dot{\hat{V}}(x(t))$ is no longer guaranteed to be rendered negative within one sampling period under the sample-and-hold implementation of $u = \Phi_{nn}(x) \in U$, in this section, another event-triggering mechanism based on errors between predicted states and measured states is developed to update the RNN models for all $x \in \Omega_{\rho_w}$. To differentiate it with the event-triggered mechanism developed for $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$ in Eq. 4.10, it will be termed the error-triggered on-line

RNN update throughout the chapter. Specifically, following the error-triggering mechanism in [5], a moving horizon error metric $E_{rnn}(t_k)$ is proposed to indicate the prediction accuracy of RNN models at $t = t_k$ as follows:

$$E_{rnn}(t_k) = \sum_{i=0}^{N_b} \frac{|x_p(t_{k-i}) - x(t_{k-i})|}{|x(t_{k-i})| + \delta} \tag{4.22}$$

where N_b is the number of sampling periods before t_k that contribute to the quantification of the prediction error. $x_p(t_{k-i})$, $i = 0, ..., N_b$ are the predictions of the past states using RNN models, while $x(t_{k-i})$ are the past state measurements from the actual nonlinear system of Eq. 4.1 under the same control actions. A small positive real number δ is added in the denominator of Eq. 4.22 to avoid the division by small numbers when $x(t_{k-i})$ approaches zero. The RNN models are updated if the following inequality is satisfied (i.e., the accumulated error $E_{rnn}(t_k)$ exceeds the threshold E_T):

$$E_{rnn}(t_k) > E_T \tag{4.23}$$

where E_T is determined via extensive closed-loop simulations. Specifically, we first choose an appropriate length N_b for the moving horizon such that it is not too short to frequently trigger the update of RNN models, nor too long to cause data-storage burden. Subsequently, based on extensive closed-loop simulations, the threshold E_T is determined off-line to trigger an RNN model update if the state error has accumulated to an undesired level. Additionally, common measurement noise (sufficiently small compared to time-varying disturbances from model uncertainty) and nonzero modeling error of RNN models should be accounted for in determining the value of E_T such that they do not trigger an update of RNN models in most times. Lastly, after the RNN model is updated, for example, at $t = r_k$, all the errors before $t = r_k$ are reset to zero.

Remark 4.5. *To ensure that an on-line update of RNN models can be accomplished within one sampling period, a new ensemble of RNN models is obtained based on previous RNN models and most recent process data. Specifically, instead of training a new ensemble of RNN models from randomly initialized weights, the weights in previous RNN models are imported as the initial weight values for the updated ensemble of RNN models. Additionally, it should be noted that only*

the initial ensemble of RNN models (i.e., pre-trained models for the nominal system of Eq. 4.1) is trained based on the entire dataset from extensive open-loop simulations. All the following updated RNN models (i.e., fine-tuning of RNN models) are developed using new collected process data.

Remark 4.6. *On-line update of RNN models via fine-tuning method (i.e., using most recent dataset only) has many advantages. First, since we initialize RNN weights which are obtained from previous RNN models, some of the underlying knowledge obtained from old datasets is transferred to the new RNN models. Additionally, by training new RNN models with the most recent dataset, the loss function in RNN learning algorithm is calculated based on the new data that captures nonlinear dynamics subject to recent disturbances only. Therefore, the updated RNN models are more capable of making accurate predictions accounting for recent disturbances. Moreover, the computation time for updating an RNN model is significantly reduced due to the small size of the newly collected dataset compared to the original training dataset. However, because of insufficient data in new training dataset, the updated RNN models are not guaranteed to approximate nonlinear dynamics subject to disturbances in the entire operating region. Therefore, RNN models will keep adapting to disturbances via the implementation of event-triggered and error-triggered mechanisms in this section, until they are accurate enough for LMPC and LEMPC to achieve closed-loop stability.*

4.3 Integration of On-line Update of RNNs with MPC

In this section, we demonstrate the implementation strategies for on-line updating RNN models in LMPC and LEMPC, respectively, following the event-triggering and error-triggering mechanisms introduced in the previous section. Subsequently, closed-loop stability is established for the nonlinear system of Eq. 4.1 subject to time-varying bounded disturbances under the sample-and-hold implementation of the LMPC of Eq. 4.8 and the LEMPC of Eq. 4.9, respectively.

4.3.1 Implementation Strategy for On-line RNN Learning Within LMPC

Based on the event-triggered and error-triggered control schemes proposed in the previous sections, the implementation strategy of the on-line RNN learning is integrated with the LMPC of Eq. 4.8 as follows:

Step 1 : An initial RNN model ensemble that is utilized in the LMPC of Eq. 4.8 is derived from extensive open-loop simulations for the nominal system of Eq. 4.1 (i.e., $w(t) \equiv 0$) following the construction method in [181].

Step 2 : Starting from an initial condition $x_0 \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$, the nonlinear system of Eq. 4.1 is operated under LMPC in a sample-and-hold fashion with states being continuously monitored and collected. The update of RNN models is triggered the moment that Eq. 4.10 is violated and the optimal control actions $u^*(t)$ will be calculated by the LMPC using the new RNN model ensemble at the next sampling time.

Step 3 : Within finite sampling periods, the closed-loop state is driven into Ω_{ρ_w} under LMPC, after which we switch to the error-triggering mechanism as discussed in the previous section. Specifically, if the current state stays in Ω_{ρ_w}, the moving horizon error detector of Eq. 4.22 and its threshold E_T are utilized to determine whether an update of RNN models is in demand. However, if the current state leaves Ω_{ρ_w} due to disturbances, the event-triggering mechanism in *Step* 2 will be re-activated to trigger an RNN model update.

Step 4 : If the closed-loop state eventually enters a small neighborhood around the origin (i.e., $\Omega_{\rho_{min}}$ defined in Eq. 2.31), which is considered to be practically stable for the nominal system of Eq. 4.1, then both the event-triggering and the error-triggering mechanisms are taken off-line until the state leaves $\Omega_{\rho_{min}}$ again. Fig. 4.1 shows a trajectory of a Lyapunov function under the LMPC with the above implementation strategy of on-line update of RNN models.

The following theorem is established to show that under the LMPC that incorporates the above implementation strategy of event-triggered on-line update of RNN models, the closed-loop state of the nonlinear system of Eq. 4.1 is bounded in the stability region $\Omega_{\hat{\rho}}$ for all times, and ultimately enters Ω_{ρ_w}. Additionally, if the disturbances in the nonlinear system of Eq. 4.1 remain unchanged

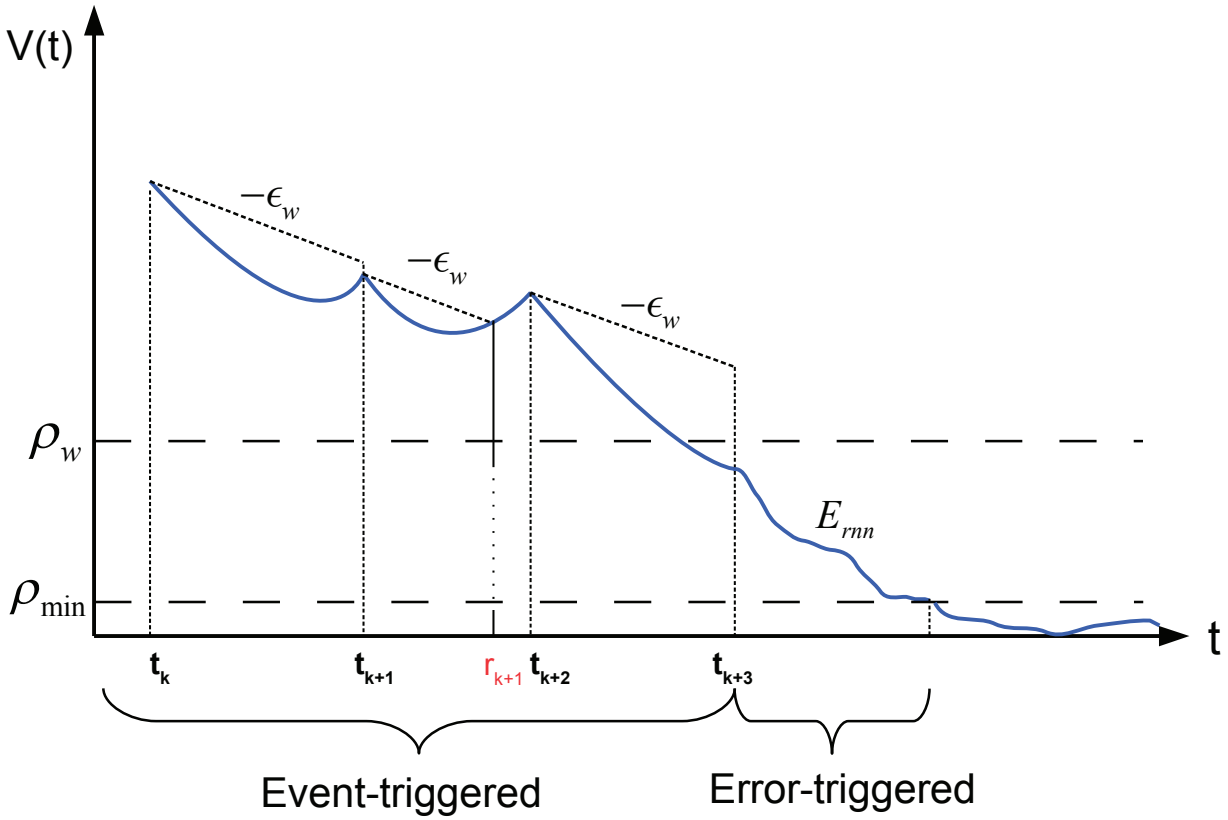


Figure 4.1: Evolution of Lyapunov function (blue trajectory) under the LMPC with event-triggered condition of Eq. 4.10 and error-triggered condition of Eq. 4.23, where the dashed lines with the slope $-\varepsilon_w$ represent the threshold lines in Eq. 4.10.

after some time, the closed-loop state can be ultimately bounded in a small neighborhood $\Omega_{\rho_{min}}$ around the origin.

Theorem 4.2. *Consider the closed-loop system of Eq. 4.1 under the LMPC of Eq. 4.8 with on-line update of RNN models. Let $\Delta > 0$, $\varepsilon_w > 0$ and $\hat{\rho} > \rho_{min} > \rho_s$ satisfy Eq. 2.30, Eq. 2.31 and Eq. 4.15. Then, given any initial state $x_0 \in \Omega_{\hat{\rho}}$, if the ensemble of RNN models is updated following the implementation strategy in this section with the triggering events of Eq. 4.10 and Eq. 4.23, then it is guaranteed that under the LMPC of Eq. 4.8, $x(t) \in \Omega_{\hat{\rho}}$, $\forall t \geq 0$, and $x(t)$ ultimately enters Ω_{ρ_w}. Additionally, if the disturbances $w(t)$ remain unchanged after $t = T_s > 0$, it holds that $\lim_{t\to\infty} \hat{V}(x(t)) \leq \rho_{min}$ for the closed-loop system of Eq. 4.1.*

Proof. We first prove that the state of the closed-loop system of Eq. 4.1 can be driven into Ω_{ρ_w} for any initial condition $x_0 \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_w}$. Since the RNN model is updated on-line following the condition in Eq. 4.10, the value of $\hat{V}(x(t))$ decreases at least at the rate of $-\varepsilon_w$ with respect to time if Eq. 4.10 is satisfied. However, in the case that an update of RNN models is triggered by the violation of Eq. 4.10 and the control actions remain unchanged until the next sampling step, it is shown in Proposition 4.2 that the state can still be driven to a smaller level set of $\hat{V}(x)$ within one sampling period. Therefore, it is guaranteed that the state ultimately converges to Ω_{ρ_w}. On the other hand, if $x(t_k) \in \Omega_{\rho_w}$, the on-line update of RNN models is subject to the error-triggering mechanism of Eq. 4.23. It is noted that the closed-loop state is not guaranteed to remain inside Ω_{ρ_w} for all times in the presence of bounded disturbances. However, once the state leaves Ω_{ρ_w}, it is shown by the characterization method of Ω_{ρ_w} that the state will not leave the closed-loop stability region $\Omega_{\hat{\rho}}$ within one sampling period, such that the state can be driven into Ω_{ρ_w} again under the LMPC of Eq. 4.8 with the event-triggering system of Eq. 4.10.

Next, we prove that after the disturbances $w(t)$ remain unchanged for all $t \geq T_s > 0$, the state of the closed-loop system of Eq. 4.1 is ultimately unbounded in ρ_{min}. Specifically, since $w(t) = w(T_s), \forall t \geq T_s$, the last updated RNN models satisfy $|v| = |F(x,u,w(T_s)) - F_{nn}(x,u)| \leq \gamma|x|$. Therefore, based on Eq. 2.30, Eq. 2.31 and Eq. 4.19, the time-derivative of $\hat{V}(x)$, $\forall t \in [t_k, t_{k+1})$,

where $t_k \geq T_s$, is bounded for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ as follows:

$$\begin{aligned}\dot{\hat{V}}(x(t)) \leq & -\hat{c}_3|x(t_k)|^2 + \hat{c}_4\gamma|x(t_k)|^2 + L_x^{'}|x(t)-x(t_k)| + L_w^{'}|w(t)-w(T_s)| \\ \leq & -\tilde{c}_3|x(t_k)|^2 + L_x^{'}M\Delta \\ \leq & -\varepsilon_w\end{aligned} \tag{4.24}$$

This implies that well-conditioned RNN models are derived to successfully capture the dynamics of nonlinear system of Eq. 4.1 in the presence of constant disturbances $w(t)$ after $t = T_s$, and therefore, the closed-loop state can be ultimately driven into Ω_{ρ_s}. Following the definitions of $\Omega_{\rho_{nn}}$, $\Omega_{\rho_{min}}$ and the proof in [181], it is demonstrated that the closed-loop state is maintained in a small neighborhood $\Omega_{\rho_{min}}$ around the origin in the presence of sufficiently small modeling error $|v|$. This completes the proof of convergence of the state to Ω_{ρ_w} within finite time, and boundedness of the state in $\Omega_{\hat{\rho}}$, $\forall t \geq 0$ for the closed-loop system of Eq. 4.1 with $x_0 \in \Omega_{\hat{\rho}}$ under the LMPC with on-line update of RNN models. □

4.3.2 Implementation Strategy for On-line RNN Learning Within LEMPC

The integrated framework of implementing on-line RNN learning within the LEMPC of Eq. 4.9 is presented as follows:

Step 1 : Similar to the implementation strategy for LMPC, an initial RNN model ensemble that is utilized in the LEMPC of Eq. 4.8 is derived from extensive open-loop simulations for the nominal system of Eq. 4.1 (i.e., $w(t) \equiv 0$) following the construction method in [171].

Step 2 : Starting from an initial condition $x_0 \in \Omega_{\hat{\rho}}$, the nonlinear system of Eq. 4.1 is operated under LEMPC in a sample-and-hold fashion with states being continuously monitored and collected. Specifically, if $x(t_k) \in \Omega_{\hat{\rho}_e}$, the RNN models are updated following the error-triggered mechanism of Eq. 4.23. However, if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$, both the event-triggered mechanism of Eq. 4.10 and the error-triggered mechanism of Eq. 4.23 are activated, where the update of RNN models is triggered by the one that violates the constraint first.

Step 3 : Since the event-triggered mechanism of Eq. 4.10 is activated for all $x \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$, the closed-loop state is guaranteed to move into $\Omega_{\hat{\rho}_e}$ within finite sampling steps. Therefore, under the time-varying operation of LEMPC with on-line updating RNNs, optimal process economic benefits and closed-loop stability are achieved simultaneously for the closed-loop system of Eq. 4.1.

The following theorem demonstrates that under the LEMPC with on-line updating RNN models, the closed-loop state of the nonlinear system of Eq. 4.1 is maintained in the stability region $\Omega_{\hat{\rho}}$ for all times.

Theorem 4.3. *Consider the closed-loop system of Eq. 4.1 under the LEMPC of Eq. 4.9 with on-line update of RNN models via the above implementation strategy. Let* $\Delta > 0$, $\varepsilon_w > 0$ *and* $\hat{\rho} > \hat{\rho}_e > \rho_w > 0$ *satisfy Eq. 2.30, Eq. 4.15 and the following inequality:*

$$\hat{\rho}_e \leq \hat{\rho} - \frac{\hat{c}_4 \sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}} f'_w(\Delta) - \kappa(f'_w(\Delta))^2 \tag{4.25}$$

where $f'_w(t) := \frac{2L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1)$. *Then, for any initial condition* $x_0 \in \Omega_{\hat{\rho}}$, *the closed-loop state* $x(t)$ *is bounded in the stability region* $\Omega_{\hat{\rho}}$, $\forall t \geq 0$.

Proof. We prove the boundedness of state in $\Omega_{\hat{\rho}}$ for the following two cases: $x(t_k) \in \Omega_{\hat{\rho}_e}$ and $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$. Specifically, we first prove that if $x(t_k) \in \Omega_{\hat{\rho}_e}$, the state of the nonlinear system of Eq. 4.1 subject to bounded disturbances does not leave $\Omega_{\hat{\rho}}$ within one sampling period (i.e., $\forall t \in [t_k, t_{k+1})$). Following the proof in [6, 175], the time-derivative of the state error vector $e(t) = x(t) - \hat{x}(t)$ is obtained $\forall x, \hat{x} \in \Omega_{\hat{\rho}}$, $u \in U$ and $w(t) \in W$ as follows:

$$\begin{aligned} |\dot{e}| =& |F(x,u,w) - F_{nn}(\hat{x},u)| \\ \leq& |F(x,u,w) - F(\hat{x},u,w(r_k)| + |F(\hat{x},u,w(r_k)) - F_{nn}(\hat{x},u)| \\ \leq& L_x |e(t)| + 2L_w w_m + \nu_m \end{aligned} \tag{4.26}$$

where it is assumed that the last updated RNN models are obtained at $t = r_k \leq t_k$. Due to the fact that measured states are fed back to the controller at every sampling step, it follows that $x(t) = \hat{x}(t)$ (i.e., $e(0) = 0$). Thus, the upper bound for $|e(t)|$ is derived for all $x(t), \hat{x}(t) \in \Omega_{\hat{\rho}}$ and $|w(t)| \leq w_m$

as follows:

$$|e(t)| = |x(t) - \hat{x}(t)| \leq \frac{2L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1) \tag{4.27}$$

Additionally, using Taylor series expansion of $\hat{V}(x)$ around $\hat{x}$ and Eq. 4.6, the following inequality is derived $\forall x, \hat{x} \in \Omega_\rho$:

$$\begin{aligned} \hat{V}(x) &\leq \hat{V}(\hat{x}) + \frac{\partial \hat{V}(\hat{x})}{\partial x}|x - \hat{x}| + \kappa |x - \hat{x}|^2 \\ &\leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa |x - \hat{x}|^2 \end{aligned} \tag{4.28}$$

where $\kappa > 0$. Therefore, from Eq. 4.28, it is demonstrated that if $\Omega_{\hat{\rho}_e}$ is characterized to satisfy Eq. 4.25, the closed-loop state $x(t)$, $t \in [t_k, t_{k+1})$ is guaranteed to be bounded in $\Omega_{\hat{\rho}}$ since the predicted state $\hat{x}(t)$ is maintained inside $\Omega_{\hat{\rho}_e}$ by the constraint of Eq. 4.8e.

On the other hand, if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$, the constraint of Eq. 4.9f is activated such that the control action u decreases the value of $\hat{V}(\hat{x})$ based on the states predicted by the RNN model of Eq. 4.9b within the next sampling period. Additionally, under the co-implementation of event-triggered mechanisms of Eq. 4.10 and Eq. 4.23, it is ensured that the state of the closed-loop system of Eq. 4.1 satisfies $\hat{V}(x(t)) < \hat{V}(x(t_k))$, $\forall t \in (t_k, t_{k+1})$, and therefore, it never leaves $\Omega_{\hat{\rho}}$, and can be eventually driven back to $\Omega_{\hat{\rho}_e}$. This completes the proof of boundedness of the closed-loop state in $\Omega_{\hat{\rho}}$ for all $x_0 \in \Omega_{\hat{\rho}}$ under LEMPC. □

Remark 4.7. *The proposed on-line update of RNN models for MPC is not limited to processes/input spaces of low dimension. Given a nonlinear system with state dimension of n, and input dimension of m, the input to the RNN model is of dimension $m+n$, and the output is of dimension n. The computational complexity of training an RNN model is approximately linear to the size of input space, and the size of each hidden layer. The computation time is not an issue for the initial RNN model since it is trained off-line based on the entire dataset. Additionally, when updating RNN models on-line, we only use the most recent data to update the RNN model instead of training a new RNN from the beginning. Therefore, the computation time is significantly reduced compared to that for the initial RNN model, and is less than one sampling period in our case. Moreover, parallel computing and hardware acceleration can be employed to further improve*

computational efficiency of training RNN models for large-scale systems.

4.4 Application to a Chemical Process Example

A chemical process example is used to illustrate the application of on-line update of RNN models for LMPC and LEMPC, respectively. Specifically, a well-mixed, non-isothermal continuous stirred tank reactor (CSTR) where an irreversible second-order exothermic reaction takes place is considered. The reaction transforms a reactant A to a product B ($A \rightarrow B$). The inlet concentration of A, the inlet temperature and feed volumetric flow rate of the reactor are C_{A0}, T_0 and F, respectively. The CSTR is equipped with a heating jacket that supplies/removes heat at a rate Q. The CSTR dynamic model is described by the following material and energy balance equations:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 \tag{4.29a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{4.29b}$$

where C_A is the concentration of reactant A in the reactor, V is the volume of the reacting liquid in the reactor, T is the temperature of the reactor and Q denotes the heat input rate. The concentration of reactant A in the feed is C_{A0}. The feed temperature and volumetric flow rate are T_0 and F, respectively. The reacting liquid has a constant density of ρ_L and a heat capacity of C_p. ΔH, k_0, E, and R represent the enthalpy of reaction, pre-exponential constant, activation energy, and ideal gas constant, respectively. Process parameter values are listed in Table 4.1.

We study the operation of CSTR under LMPC and LEMPC with the same unstable steady-state $(C_{As},\ T_s) = (1.95\ kmol/m^3,\ 402\ K)$, and $(C_{A0_s}\ Q_s) = (4\ kmol/m^3,\ 0\ kJ/hr)$. The manipulated inputs are the inlet concentration of species A and the heat input rate, which are represented by the deviation variables $\Delta C_{A0} = C_{A0} - C_{A0_s}$, $\Delta Q = Q - Q_s$, respectively. The manipulated inputs are bounded as follows: $|\Delta C_{A0}| \leq 3.5\ kmol/m^3$ and $|\Delta Q| \leq 5 \times 10^5\ kJ/hr$. Therefore, the states and the inputs of the closed-loop system are $x^T = [C_A - C_{As}\ T - T_s]$ and $u^T = [\Delta C_{A0}\ \Delta Q]$, respectively, such

Table 4.1: Parameter values of the CSTR with a second-order reaction.

$T_0 = 300\ K$	$F = 5\ m^3/hr$
$V = 1\ m^3$	$E = 5 \times 10^4\ kJ/kmol$
$k_0 = 8.46 \times 10^6\ m^3/kmol\ hr$	$\Delta H = -1.15 \times 10^4\ kJ/kmol$
$C_p = 0.231\ kJ/kg\ K$	$R = 8.314\ kJ/kmol\ K$
$\rho_L = 1000\ kg/m^3$	$C_{A0_s} = 4\ kmol/m^3$
$Q_s = 0.0\ kJ/hr$	$C_{A_s} = 1.95\ kmol/m^3$
$T_s = 402\ K$	

that the equilibrium point of the system is at the origin of the state-space, (i.e., $(x_s^*, u_s^*) = (0,0)$). In this study, we consider the model variations caused by the following disturbances. 1) The feed flow rate F is varying due to an upstream disturbance that F becomes time-varying with the constraint: $0 \leq F \leq 12\ m^3/hr$. 2) Additionally, catalyst activation is accounted for during the operation of the CSTR of Eq. 4.29, which leads to a reduction in the reaction pre-exponential factor k_0 with the constraint: $0 < k_0 < 8.46 \times 10^6\ m^3/kmol\ hr$.

The control Lyapunov function $V(x) = x^T P x$ is designed with the following positive definite P matrix:

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{4.30}$$

Then, the closed-loop stability region Ω_ρ for the CSTR is characterized as a level set of the Lyapunov function with $\hat{\rho} = 368$ inside the region ϕ_u, from which the origin can be rendered exponentially stable under the controller $u = \Phi(x) \in U$.

The explicit Euler method with an integration time step of $h_c = 10^{-4}\ hr$ is used to numerically simulate the dynamic model of Eq. 4.29. The nonlinear optimization problem of the LMPC of Eq. 4.8 is solved using the python module of the IPOPT software package [158], named PyIpopt with the sampling period $\Delta = 10^{-2}\ hr$. The initial ensemble of RNN models is generated following the data generation and RNN learning algorithm in [171, 181]. Parallel computing is employed to carry out the calculations of multiple RNN predictions concurrently using a Message Passing

Interface (MPI) for the Python programming language, named *MPI4Py* ([40]).

In this study, we assume that there is no noise in real-time data. However, in the case where the state measurements are noisy, the proposed on-line update of RNN models can still be applied via a data pre-processing step to smoothen the measurement data. For example, Savitzky-Golay filter, a generalized moving average based on the least squares fitting, can be applied to smoothen noisy sampled data without distorting the data tendency before feeding real-time data into the RNN model [108, 131].

Closed-loop Simulation under LMPC

The control objective of LMPC is to operate the CSTR at the unstable equilibrium point $(C_{As},\ T_s)$ by manipulating the heat input rate ΔQ and the inlet concentration ΔC_{A0} under the LMPC using RNN models. The closed-loop simulation results for the nominal system of Eq. 4.29 under LMPC are shown in [181], where it is demonstrated that the state converges to a small neighborhood $\Omega_{\rho_{min}}$ around the origin ultimately. The simulation results for the uncertain system of Eq. 4.29 under LMPC with on-line update of RNN model ensemble are shown in Figs. 4.2-4.8. Specifically, the feed flow rate F is increased to 12 m^3/hr at $t = 0.05$ hr, and k_0 is gradually decreased to $0.8k_0$, $0.6k_0$ and $0.4k_0$ at $t = 0.1$ hr, 0.2 hr and 0.4 hr, respectively, and remains unchanged afterwards. In Fig. 4.2, it is shown that the closed-loop state trajectory under LMPC without on-line update of RNN model ensemble (i.e., using the initial RNN model ensemble for all times) oscillates around the origin due to disturbances, while the LMPC with on-line update of RNN model ensemble successfully drives the closed-loop state into a small neighborhood around the origin. Additionally, in Fig. 4.3 and Fig. 4.4, it is shown that the closed-loop states under the LMPC with on-line RNN update are stabilized at their steady-states after $t = 0.2$ hr, while those under the LMPC without on-line RNN update shows considerable oscillation since the initial RNN model ensemble is not able to capture dynamic behavior of the system of Eq. 4.29 in the presence of disturbances. Therefore, the dynamic performance of the closed-loop system of Eq. 4.29 under the LMPC is significantly improved through on-line update of RNN model ensemble.

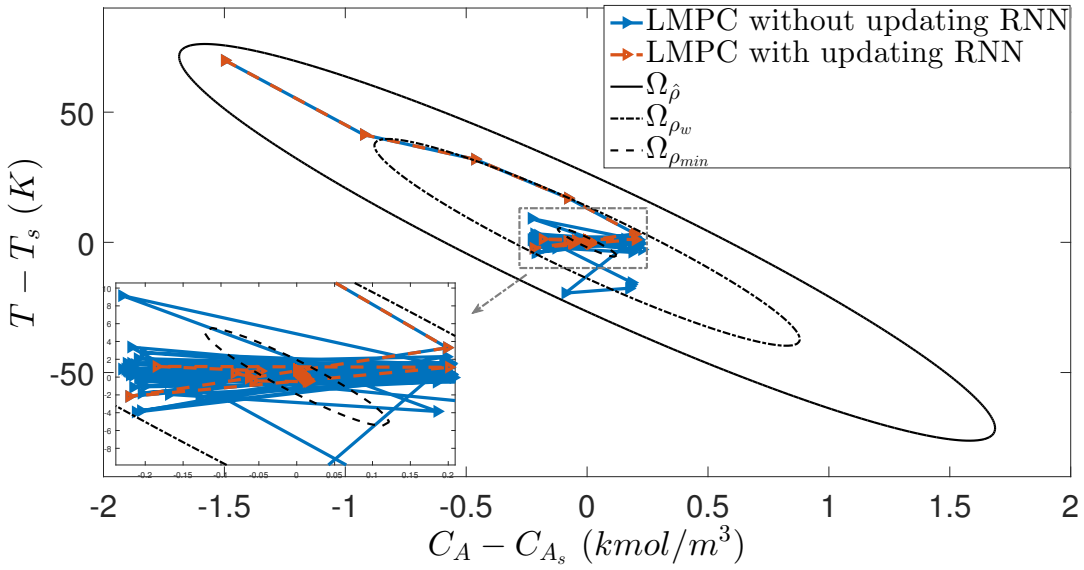


Figure 4.2: The state-space profiles for the closed-loop CSTR under the LMPC of Eq. 4.8 with and without on-line update of RNN model ensemble for the initial condition (-1.5, 70).

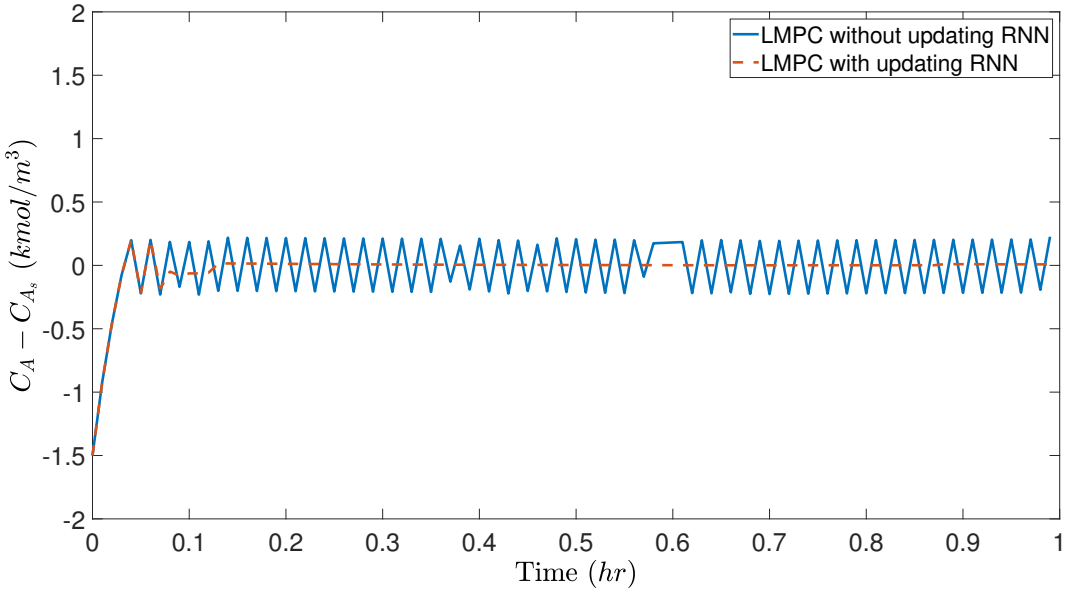


Figure 4.3: The state profiles ($x_1 = C_A - C_{As}$) for the initial condition (-1.5, 70) under the LMPC of Eq. 4.8 with and without on-line update of RNN model ensemble, respectively.

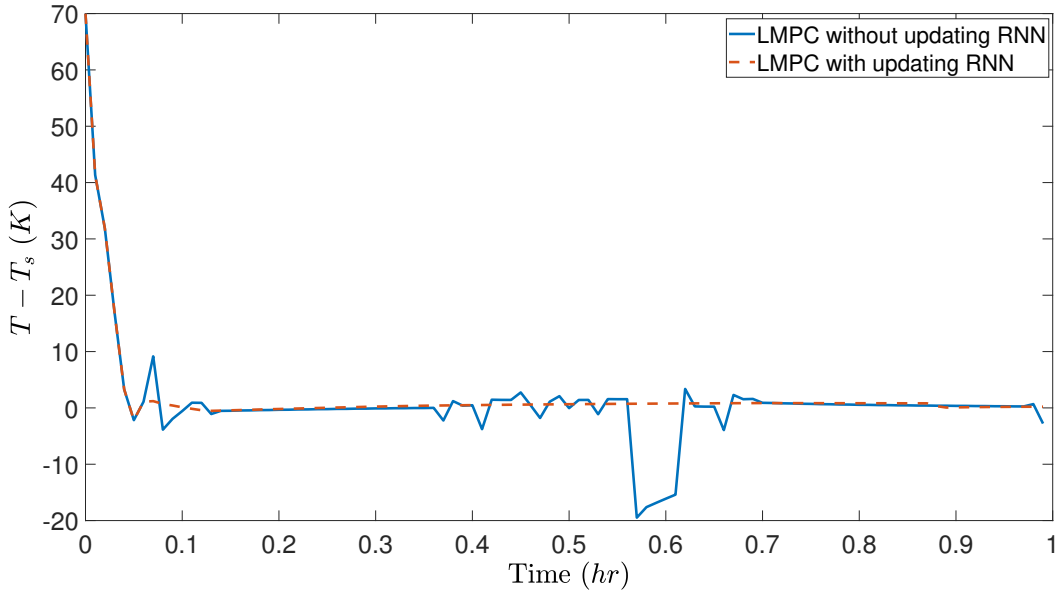


Figure 4.4: The state profiles ($x_2 = T - T_s$) for the initial condition (-1.5, 70) under the LMPC of Eq. 4.8 with and without on-line update of RNN model ensemble, respectively.

Fig. 4.5 shows the evolution of moving horizon error detector $E_{rnn}(t)$ for the closed-loop system of Eq. 4.29 under the LMPC of Eq. 4.8 with error-triggered on-line update of RNN models. Specifically, since it takes only one sampling step for the closed-loop state to enter Ω_{ρ_w}, the event-triggering condition of Eq. 4.10 is never triggered in this case. Additionally, in Fig. 4.5, it is shown that the update of RNN models is triggered two times with the threshold $E_T = 15$. After the closed-loop state enters a small neighborhood around the origin (i.e., $\Omega_{\rho_{\min}}$), the error-triggering system is off-line according to the implementation strategy (i.e., *Step* 4) for LMPC.

Fig. 4.6 depicts the evolution of the Lyapunov function value, $\hat{V}(x)$, of the closed-loop state, under the LMPC with and without on-line update of RNN models, respectively. In Fig. 4.6, the closed-loop state under on-line update enters $\Omega_{\rho_{min}}$ after $t = 0.1$ hr in the presence of disturbances, while it oscillates heavily and never enters $\Omega_{\rho_{min}}$ under the LMPC without on-line model update. Finally, in Fig. 4.7 and Fig. 4.8, the manipulated input profiles for $u_1 = \Delta C_{A0}$ and $u_2 = \Delta Q$ are presented for both LMPCs. Specifically, when the RNN models are updated on-line, u_1 in Fig. 4.7 settles to its steady-state value after $t = 0.12$ hr. However, without on-line update of RNN models, u_1 shows sustained oscillation between the maximum and minimum saturated points, which might

significantly shorten the lifespan of actuators. Similarly, in Fig. 4.8, the LMPC with on-line update of RNN models shows smoother control actions u_2 compared to that without on-line model update.

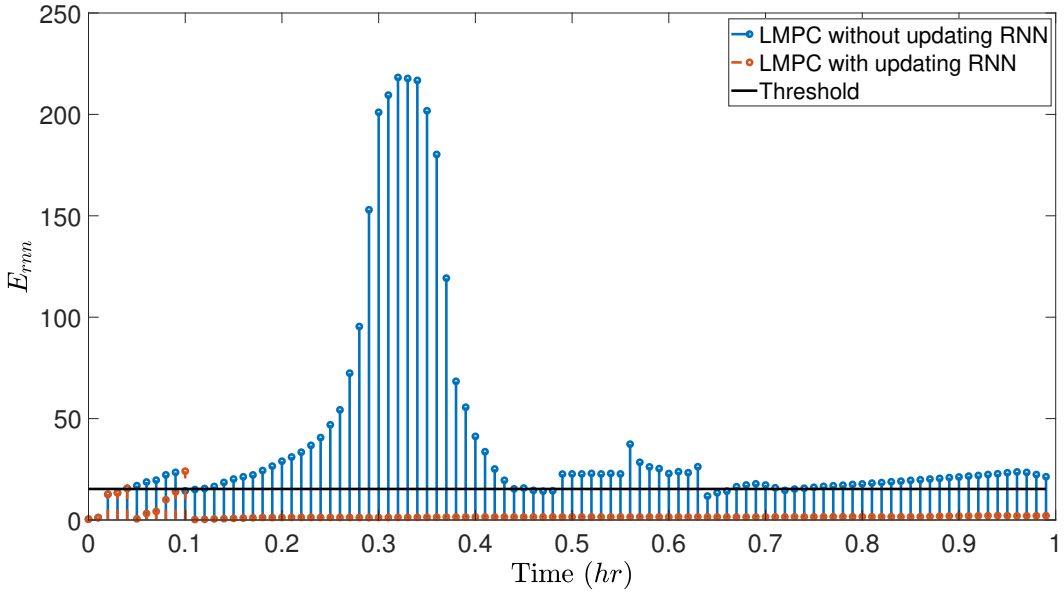


Figure 4.5: Value of $E_{rnn}(t)$ of Eq. 4.22 at each sampling time for the closed-loop system of Eq. 4.29 under the LMPC of Eq. 4.8 with error-triggered on-line update of RNN models.

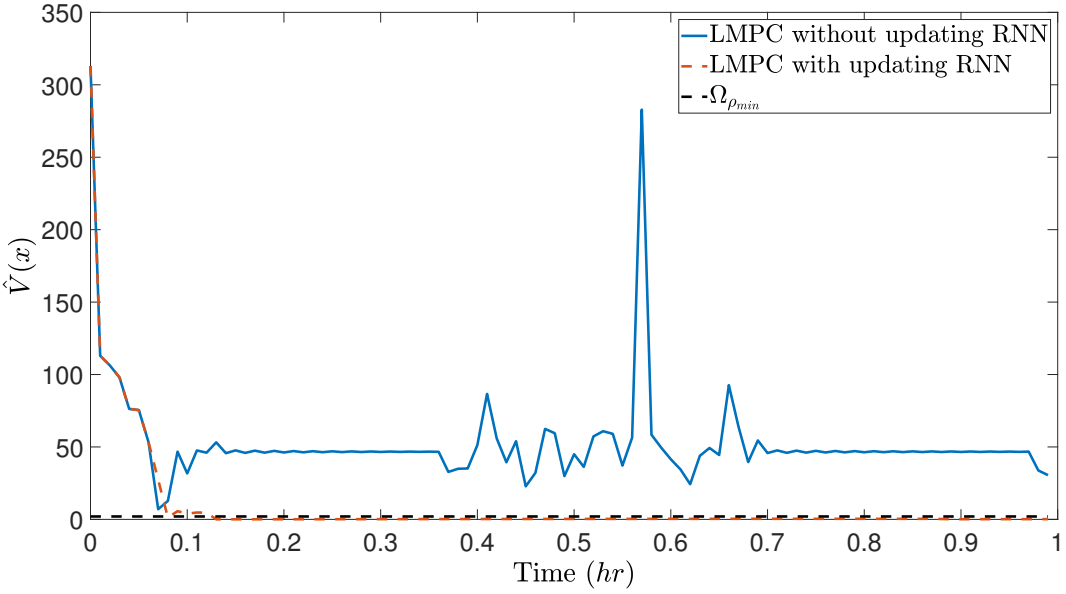


Figure 4.6: Evolution of $\hat{V}(x)$ for the closed-loop system of Eq. 4.29 under the LMPC of Eq. 4.8 with and without error-triggered on-line update of RNN models.

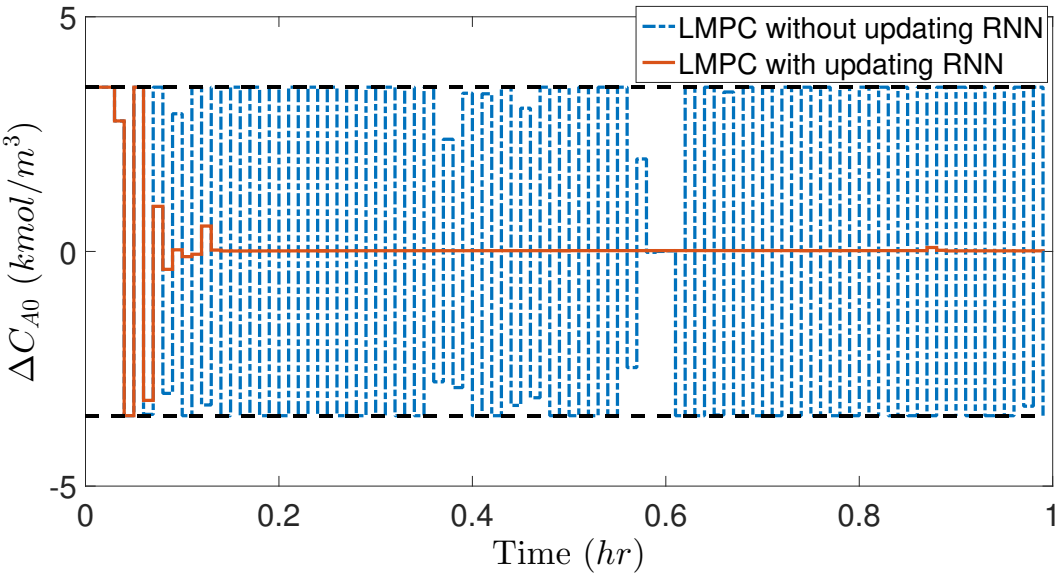


Figure 4.7: Manipulated input profiles ($u_1 = \Delta C_{A0}$) for the initial condition (-1.5, 70) under the LMPC of Eq. 4.8 with and without on-line update of RNN model ensemble, respectively, where the black dotted lines represent the upper and lower bound for u_1.

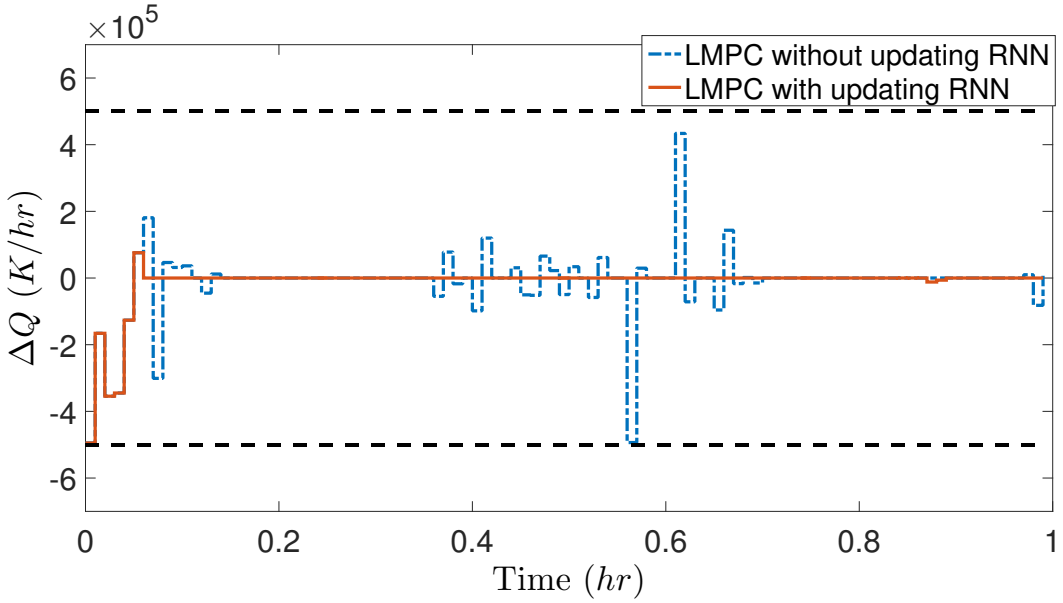


Figure 4.8: Manipulated input profiles ($u_2 = \Delta Q$) for the initial condition (-1.5, 70) under the LMPC of Eq. 4.8 with and without on-line update of RNN model ensemble, respectively, where the black dotted lines represent the upper and lower bound for u_2.

Closed-loop Simulation under LEMPC

The control objective of LEMPC is to maximize the profit of the CSTR process of Eq. 4.29 by manipulating the inlet concentration ΔC_{A0} and the heat input rate ΔQ, and meanwhile maintain the closed-loop state trajectories in the stability region $\Omega_{\hat{\rho}}$ for all times under LEMPC. The objective function of the LEMPC optimizes the production rate of B as follows:

$$l_e(\tilde{x}, u) = k_0 e^{-E/RT} C_A^2 \tag{4.31}$$

Additionally, the following material constraint is utilized in the LEMPC of Eq. 4.9 to make the averaged reactant material available within the operating period t_p to be its steady-state value, C_{A0s} (i.e., the averaged reactant material in deviation form, u_1, is equal to 0).

$$\frac{1}{t_p}\int_0^{t_p} u_1(\tau) d\tau = 0 \; kmol/m^3 \tag{4.32}$$

In [171], it has been demonstrated that the closed-loop state of the nominal system of Eq. 4.29 is bounded in $\Omega_{\hat{\rho}}$ for all times under LEMPC. In this study, we consider the same disturbances that we have performed for the closed-loop system of Eq. 4.29 under LMPC. Additionally, the CSTR system of Eq. 4.29 is operated under LEMPC for five consecutive operation periods with $t_p = 0.2 \; hr$ for each operation period. The simulations results for the closed-loop system of Eq. 4.29 in the presence of disturbances are shown in Figs. 4.9-4.15. Specifically, in Fig. 4.9, it is shown that the closed-loop state circles inside the stability region $\Omega_{\hat{\rho}}$ due to the time-varying operation under LEMPC. Additionally, it is demonstrated that the closed-loop state is bounded in $\Omega_{\hat{\rho}}$ for all times under the LEMPC of Eq. 4.9 with on-line update of RNN models. From Fig. 4.10, it is shown that the moving horizon error detector E_{rnn} exceeds the threshold twice under the LEMPC with on-line update of RNN models (i.e., the RNN update is triggered twice), and ultimately remains at a low value (below the threshold) after a more accurate ensemble of RNN models are derived to account for process disturbances. However, it is observed that the error detector E_{rnn} under the

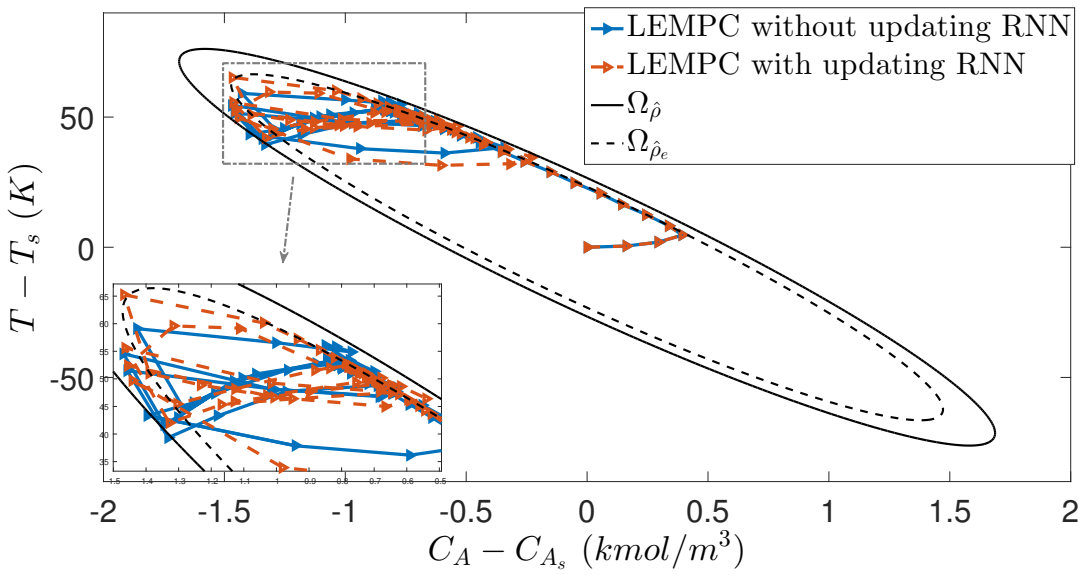


Figure 4.9: The state trajectories for the closed-loop CSTR under the LEMPC of Eq. 4.9 with and without on-line update of RNN model ensemble for the initial condition (0, 0).

LEMPC without on-line update of RNN models maintains at a high level (close to the threshold) for all times, which implies that the deviation between the predicted state and the actual states is considerable, and may lead to undesired closed-loop performance. Moreover, it is observed in Fig. 4.13 that the event-triggered mechanism of Eq. 4.10 is never activated in this case since $\hat{V}(x)$ decreases rapidly for all states outside $\Omega_{\hat{\rho}_e}$, and thus, satisfies $\hat{V}(x(t)) \leq \hat{V}(x(t_k)) - \varepsilon_w(t-t_k)$, $t \in [t_k, t_{k+1})$.

Based on the state profiles shown in Fig. 4.11 and Fig. 4.12, the evolution of the value of $\hat{V}(x)$ for the closed-loop system of Eq. 4.29 is compared between the LEMPC with and without on-line update of RNN models in Fig. 4.13. Specifically, it is shown that $\hat{V}(x)$ under LEMPC with on-line update of RNN models remains below 368 (i.e., the value of $\hat{\rho}$ for the closed-loop stability region $\Omega_{\hat{\rho}}$) for all times, while it exceeds 368 under the LEMPC without on-line update of RNN models around $t = 0.2$ *hr* and $t = 0.6$ *hr*. Additionally, since the accuracy of RNN prediction for nonlinear dynamics of Eq. 4.29 subject to disturbances is improved via on-line update using real-time process data, $\hat{V}(x)$ is smoothly maintained below $\hat{\rho}_e$ during the last 0.4 *hr*. However, $\hat{V}(x)$ based on the states under the LEMPC without on-line update of RNN models shows sustained oscillation around

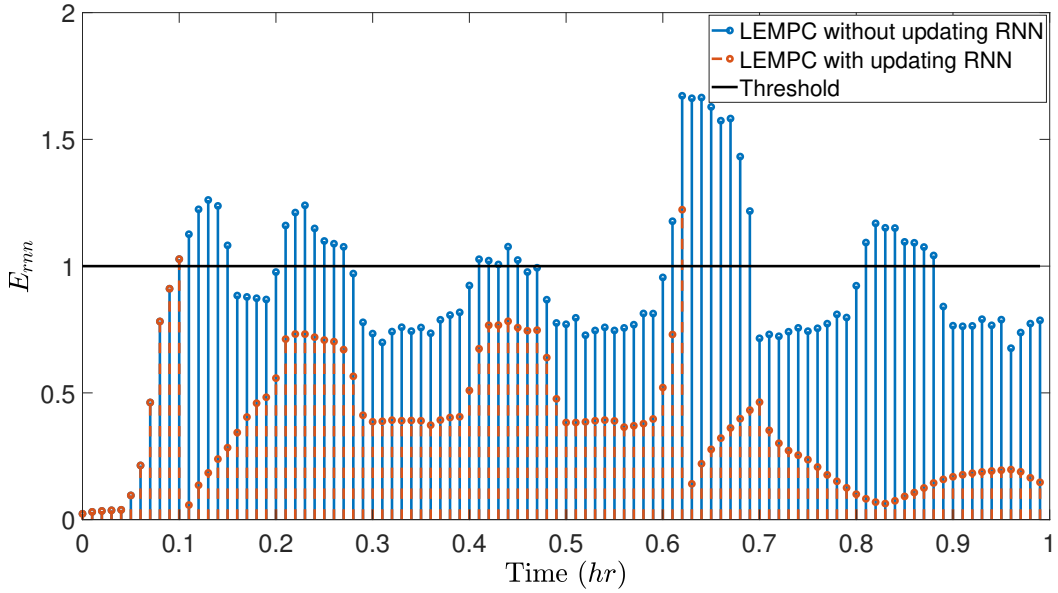


Figure 4.10: Value of $E_{rnn}(t)$ of Eq. 4.22 at each sampling time for the closed-loop system of Eq. 4.29 under the LEMPC of Eq. 4.9 with error-triggered on-line update of RNN models.

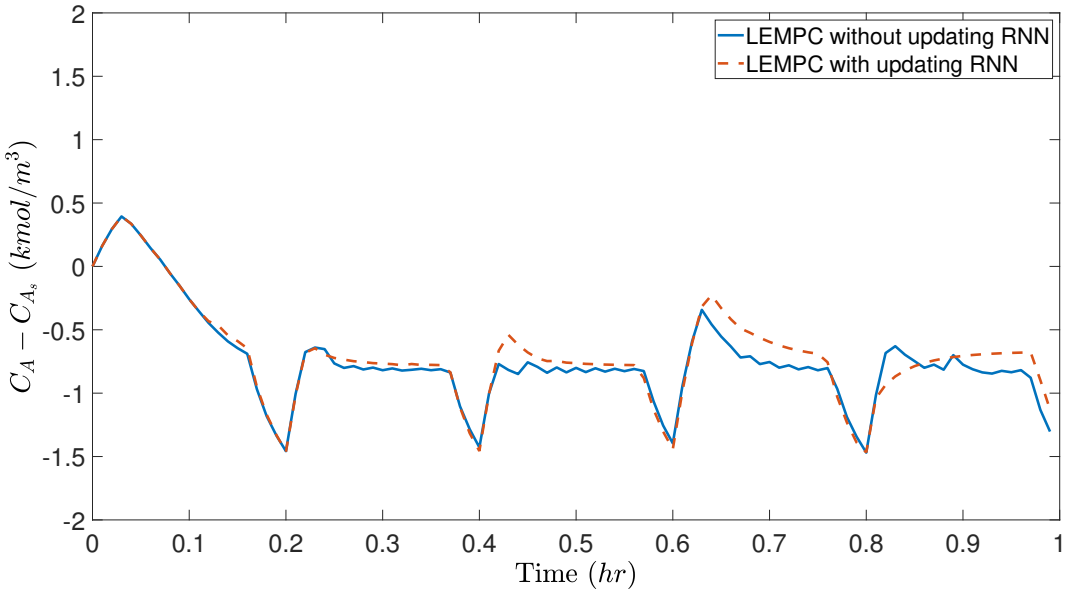


Figure 4.11: The state profiles ($x_1 = C_A - C_{As}$) for the initial condition (0, 0) under the LEMPC of Eq. 4.9 with and without on-line update of RNN model ensemble, respectively.

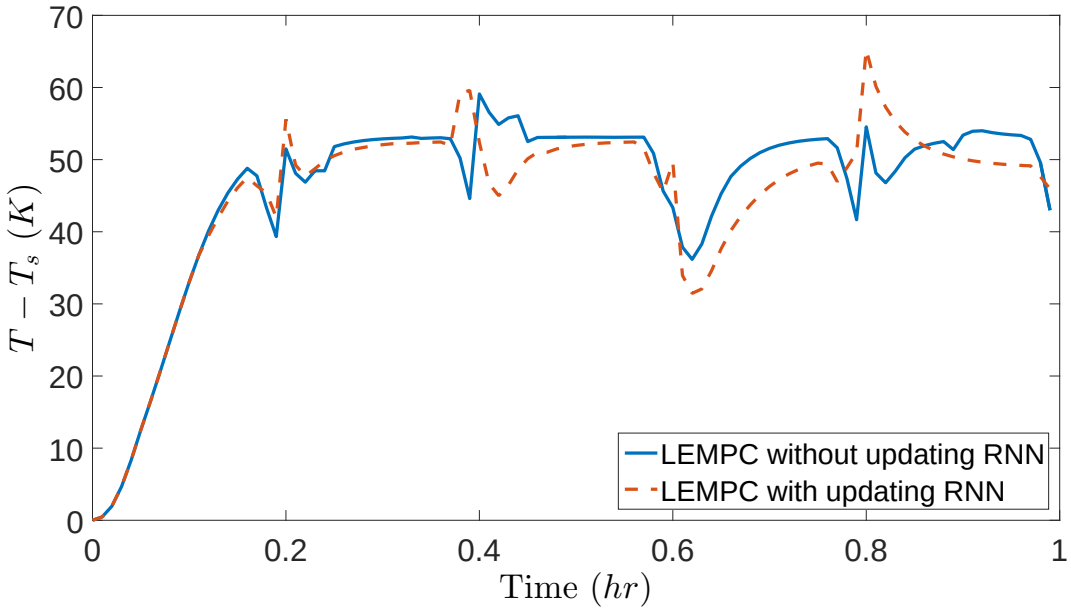


Figure 4.12: The state profiles ($x_2 = T - T_s$) for the initial condition (0, 0) under the LEMPC of Eq. 4.9 with and without on-line update of RNN model ensemble, respectively.

$\hat{\rho}_e$ due to significant model mismatch as indicated in Fig. 4.10.

Manipulated input profiles for the closed-loop system of Eq. 4.29 are given in Fig. 4.14 and Fig. 4.15, in which it is shown that the input constraints on ΔC_{A0} and ΔQ are satisfied for all times. Additionally, it is observed in Fig. 4.14 that the closed-loop system initially consumes the maximum allowable ΔC_{A0} (i.e., $\Delta C_{A0} = 3.5 kmol/m^3$) within each operation period ($t_p = 0.2\ hr$) to maximize the production rate of B, and therefore, has to lower the reactant consumption near the end of each operation period to meet the material constraint of Eq. 4.32 for all times.

Lastly, the total economic benefits achieved within five operation periods are calculated for the LEMPC with on-line update of RNN models and the steady-state operation (i.e., the system of Eq. 4.29 is operated at $(C_{As},\ T_s)$ for all times) using the following equation:

$$L_E = \int_0^{5t_p} l_e(x,u)dt$$

It is shown that $L_E = 16.74$ for the closed-loop system under LEMPC and $L_E = 10.23$ for the steady-state operation within 1 hr. Therefore, it is concluded that time-varying operation of the

system of Eq. 4.29 under the LEMPC of Eq. 4.9 with on-line updating RNN models achieves higher economic benefits compared to the steady-state operation, and outperforms that without on-line update of RNN models in terms of smoother operation and stronger robustness properties.

Remark 4.8. *The quasi-periodicity of closed-loop state and input profiles is due to the reactant material constraint that is incorporated in LEMPC. Since it is required that the averaged reactant material used within each operating period is equal to its steady-state value (i.e., reactant material constraint), LEMPC consumes the maximum allowable reactants and energy at the early stage of each operating period (owing to the second-order reaction rate to maximize reaction rate), and lowers the reactant consumption near the end of the period to meet the material constraint. In the simulation, the CSTR system is operated under LEMPC for five operating periods, and therefore, the state and input profiles exhibit quasi-periodic behavior.*

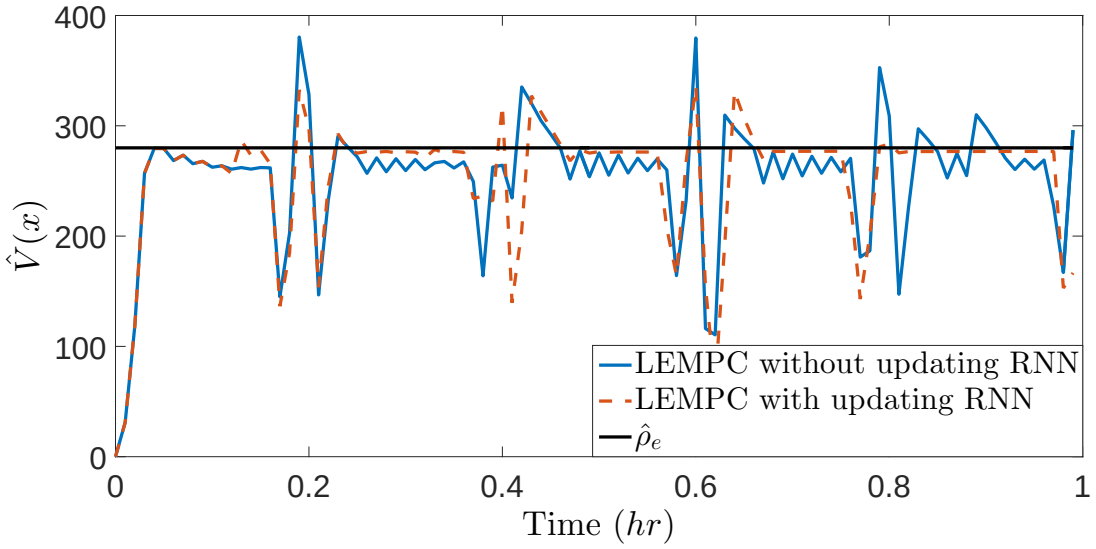


Figure 4.13: Evolution of $\hat{V}(x)$ for the closed-loop system of Eq. 4.29 under the LEMPC of Eq. 4.9 with and without error-triggered on-line update of RNN models, respectively.

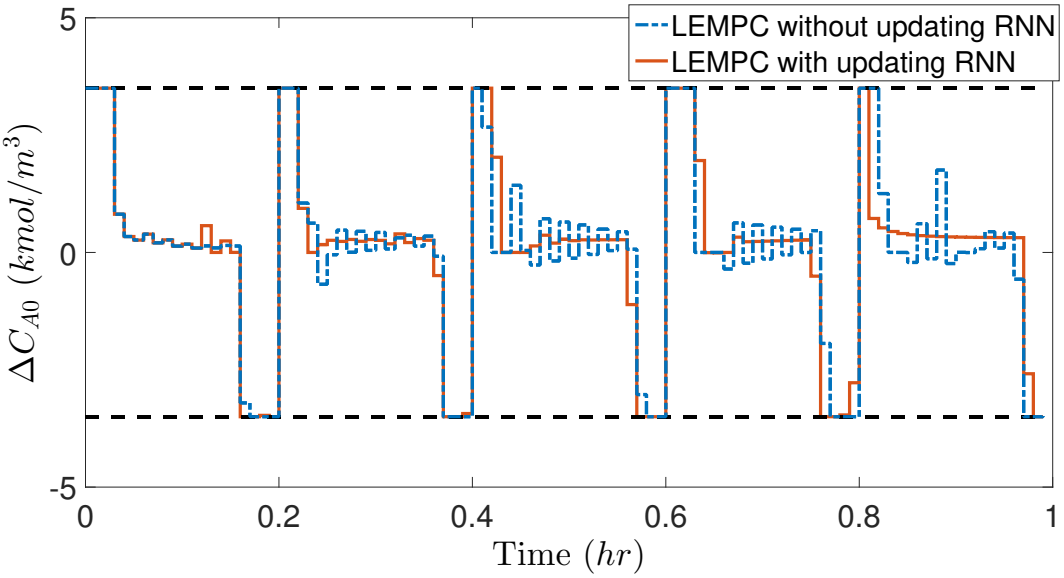


Figure 4.14: Manipulated input profiles ($u_1 = \Delta C_{A0}$) for the initial condition (0, 0) under the LEMPC of Eq. 4.9 with and without on-line update of RNN model ensemble, respectively, where the black dotted lines represent the upper and lower bounds for u_1.

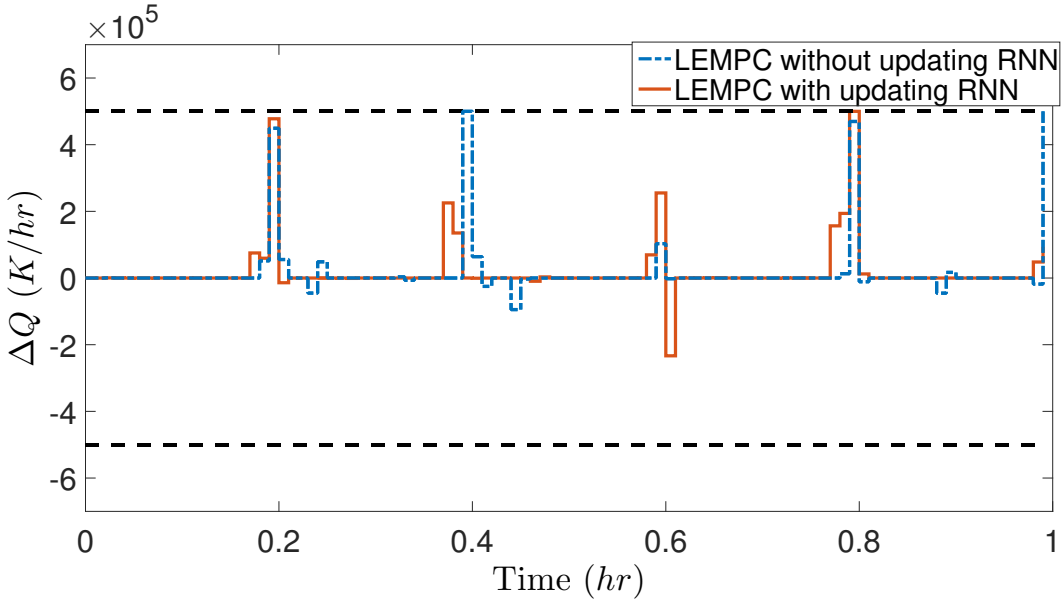


Figure 4.15: Manipulated input profiles ($u_2 = \Delta Q$) for the initial condition (0, 0) under the LEMPC of Eq. 4.9 with and without on-line update of RNN model ensemble, respectively, where the black dotted lines represent the upper and lower bounds for u_2.

4.5 Conclusion

This chapter focused on the real-time implementation of machine learning-based MPC and EMPC to nonlinear processes subject to time-varying disturbances. Based on the ensemble of RNN models that were obtained from extensive simulation data, Lyapunov-based MPC was developed to drive the state of the nominal closed-loop system to the steady-state, and Lyapunov-based EMPC was developed to maintain the state in the closed-loop stability region, respectively. Subsequently, event-triggered and error-triggered mechanisms were incorporated in LMPC and LEMPC to update the RNN models on-line using the most recent process data that account for nonlinear dynamics in the presence of disturbances. The application of the proposed methodology to a chemical process example demonstrated that the closed-loop state converged to the origin under LMPC, and remained bounded in the closed-loop stability region under LEMPC with improved dynamic performance compared to those without on-line update of RNN models.

Chapter 5

Physics-Based Machine Learning Modeling of Nonlinear Systems

5.1 Introduction

Recurrent neural networks (RNN), a class of artificial neural networks that can represent temporal dynamic behavior through feedback loops in neurons, have been utilized to model nonlinear dynamic systems and have been incorporated in the design of model predictive controllers (MPC) that optimize process performance based on RNN prediction results in Chapter 2. However, as neural network modeling is generally treated as a black-box modeling approach where no physical knowledge is utilized, interpretability and optimality of neural network modeling remain questionable. On the other hand, chemical processes have been studied for a long time by researchers and engineers, where first-principles knowledge has been obtained based on their predefined and well-known structure. For example, a chemical plant is designed in a sequence of intricate operation units that perform reactions, separations, among many others operations in which raw materials are fed in the first unit and products are obtained in the last unit in its simplest structure. Additionally, it is also very common that some processes are highly coupled among units through reflux of unreacted material that is recycled to upstream units to maximize

the production [88, 150]. However, at this stage, the incorporation of first-principles or physical knowledge of chemical processes into RNN modeling has not been thoroughly studied.

Fully-connected neural networks are developed based on the assumption that all the inputs affect all the neural network neurons, followed by all the outputs. However, it is noted that in realistic chemical processes, it is common that only a portion of inputs affect a portion of outputs, for example, in a multiple unit process in which upstream units affect downstream units but not in the opposite direction. In order to make better use of such a priori process knowledge, many researchers have started to incorporate physical knowledge of systems in the neural network formulation (e.g., [15, 70, 71, 86, 87, 138]). For example, hybrid models and gray-box models have been developed to introduce chemical process knowledge into data-driven modeling in early works [45, 50, 68, 69, 123, 143, 146, 183, 193].

Motivated by the above considerations, in this chapter, we develop a hybrid model, a partially-connected RNN model, and a weight-constrained RNN model to incorporate process physical knowledge into RNN modeling and training. Subsequently, the proposed partially-connected RNN model and the weight-constrained RNN model are incorporated in the design of MPC and of economic MPC (EMPC) to provide predictions of future states for the optimization problem of MPC and EMPC that optimize process performance in terms of closed-loop stability and economic optimality. Finally, the RNN-MPC and RNN-EMPC are applied to a chemical process example to demonstrate their improved closed-loop performances in terms of faster convergence to the steady-state under RNN-MPC and enhanced process economic profits under RNN-EMPC than the controllers using a fully-connected RNN model.

5.1.1 Notation

The Euclidean norm of a vector is denoted by the operator $|\cdot|$ and the weighted Euclidean norm of a vector is denoted by the operator $|\cdot|_Q$ where Q is a positive definite matrix. x^T denotes the transpose of x. The notation $L_f V(x)$ denotes the standard Lie derivative $L_f V(x) := \frac{\partial V(x)}{\partial x} f(x)$. Set subtraction is denoted by "\", i.e., $A \backslash B := \{x \in \mathbf{R}^n \mid x \in A, x \notin B\}$.

5.1.2 Class of Systems

The class of continuous-time nonlinear systems considered is described by the following state-space form:

$$\dot{x} = F(x,u,w) := f(x) + g(x)u + h(x)w,\ x(t_0) = x_0 \tag{5.1}$$

where $x \in \mathbf{R}^n$ is the state vector, $u \in \mathbf{R}^m$ is the manipulated input vector, and $w \in \mathbf{W}$ is the disturbance vector, where $\mathbf{W} := \{w \in \mathbf{R}^l \mid |w| \leq \theta,\ \theta \geq 0\}$. The control action constraint is defined by $u \in U := \{u_{\min} \leq u \leq u_{\max}\} \subset \mathbf{R}^m$, where u_{min} and u_{max} represent the minimum and the maximum value vectors of inputs allowed, respectively. $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are sufficiently smooth vector and matrix functions of dimensions $n \times 1$, $n \times m$, and $n \times l$, respectively. Without loss of generality, the initial time t_0 is taken to be zero ($t_0 = 0$), and it is assumed that $f(0) = 0$, and thus, the origin is a steady-state of the system of Eq. 5.1 with $u(t) = w(t) \equiv 0$.

5.1.3 Stabilizability Assumptions Expressed via Lyapunov-based Control

We assume that there exists a positive definite and proper Control Lyapunov function (CLF) V for the nominal system of Eq. 5.1 with $w(t) \equiv 0$ that satisfies the small control property (i.e., for every $\varepsilon > 0$, $\exists\ \delta > 0$, s.t. $\forall\ x \in \mathscr{B}_\delta(0)$, there exists u that satisfies $|u| < \varepsilon$ and $L_f V(x) + L_g V(x)u < 0$, [142]) and the following condition:

$$L_f V(x) < 0, \forall x \in \{z \in \mathbf{R}^n \backslash \{0\} \mid L_g V(z) = 0\} \tag{5.2}$$

The CLF assumption implies that there exists a stabilizing feedback control law $\Phi(x) \in U$ for the nominal system of Eq. 5.1 (i.e., $w(t) \equiv 0$) that renders the origin of the closed-loop system asymptotically stable for all x in a neighborhood of the origin in the sense that $L_f V(x) + L_g V(x)u < 0$ holds for $u = \Phi(x) \in U$. An example of a feedback control law can be found in [83]. Based on the CLF assumption, we can first characterize a region where the time-derivative of V is rendered negative definite under the controller $\Phi(x) \in U$ as $\phi_u = \{x \in \mathbf{R}^n \mid \dot{V}(x) = L_f V + L_g V u <$

$-kV(x), u = \Phi(x) \in U\} \cup \{0\}$, where k is a positive real number. Then, the closed-loop stability region Ω_ρ for the nonlinear system of Eq. 5.1 is defined as a level set of the Lyapunov function embedded in ϕ_u: $\Omega_\rho := \{x \in \phi_u \mid V(x) \leq \rho\} \subset \phi_u$, where $\rho > 0$.

Remark 5.1. *We consider the nonlinear system with the form of Eq. 5.1 since control-affine nonlinear systems are very common in the modeling of chemical processes. Additionally, with the form of Eq. 5.1 , we can simplify the discussion on the design of a stabilizing controller $u = \Phi(x)$ by using the Sontag control law [83]. However, it should be noted that the proposed RNN modeling approaches that account for a priori process knowledge in this chapter are not restricted to control-affine nonlinear systems, and can be generalized to nonlinear systems in a more general form: $\dot{x} = f(x,u,w)$.*

5.1.4 Recurrent Neural Network Model

A recurrent neural network (RNN) model that approximates the nonlinear dynamics of the system of Eq. 5.1 is developed with the following form:

$$\dot{\hat{x}} = F_{nn}(\hat{x}, u) := A\hat{x} + \Theta^T y \tag{5.3}$$

where $\hat{x} \in \mathbf{R}^n$ is the RNN state vector and $u \in \mathbf{R}^m$ is the manipulated input vector. $y = [y_1, ..., y_n, y_{n+1}, ..., y_{m+n}] = [\sigma(\hat{x}_1), ..., \sigma(\hat{x}_n), u_1, ..., u_m] \in \mathbf{R}^{n+m}$ is a vector of both the network state $\hat{x}$ and the input u, where $\sigma(\cdot)$ is the nonlinear activation function (e.g., a sigmoid function $\sigma(x) = 1/(1+e^{-x})$). A is a diagonal coefficient matrix, i.e., $A = diag\{-a_1, ..., -a_n\} \in \mathbf{R}^{n \times n}$, and $\Theta = [\theta_1, ..., \theta_n] \in \mathbf{R}^{(m+n) \times n}$ with $\theta_i = b_i[w_{i1}, ..., w_{i(m+n)}]$, $i = 1, ..., n$. a_i and b_i are constants. w_{ij} is the weight connecting the jth input to the ith neuron where $i = 1, ..., n$ and $j = 1, ..., (m+n)$. a_i is assumed to be positive such that each state $\hat{x}_i$ is bounded-input bounded-state stable. It is noted that to simplify the mathematical expressions of the input vector, and of the weight matrix in this chapter, we do not include the bias term in the notation since it can always be considered as an additional constant input (i.e., $u \in \mathbf{R}^{m+1}$), and therefore, does not affect the formulation of the

continuous RNN models.

Although the universal approximation theorem states that a neural network with a single hidden layer with sufficient number of neurons can approximate any continuous-time nonlinear function on compact subsets of $\mathbf{R}^n$, algorithmic learnability of the optimal neural network weights is not guaranteed. In fact, due to the complexity of neural network structure, availability of computing power, and feasibility of optimization algorithms, it is challenging to find such an optimal weight for the regression problems of a large-scale, complex system. Therefore, how to improve the performance of neural networks has been a major long-standing challenge for researchers in machine learning community over the past few decades, where a lot of efforts have been made to optimize neural network structure, develop advanced optimization algorithms, improve data-processing systems and so on.

In this chapter, we will improve the performance of RNN models in terms of enhanced prediction accuracy by incorporating structural domain knowledge of the nonlinear system of Eq. 5.1 (i.e., knowledge of the dependence of the state variables) into the development of the RNN structure. Specifically, instead of treating the RNN system of Eq. 5.3 like a black box and training it using all the inputs and outputs available (termed the fully-connected model throughout the chapter), we modify the RNN structure according to the structural process knowledge of the nonlinear system of Eq. 5.1. The details of the proposed new structure are discussed in the following section.

Remark 5.2. *It is noted that in Eq. 5.3, we use a one-hidden-layer RNN model with n states in order to simplify the discussion of the approximation of the nonlinear system of Eq. 5.1 using an RNN model. However, the RNN modeling method in this section is not restricted to a one-hidden layer RNN structure with n states only. The RNN states $\hat{x} \in \mathbf{R}^n$ in Eq. 5.3 can be considered to be the last hidden layer (if the output of the nonlinear system of Eq. 5.1 is a function of states), or the output layer of an RNN (if the state x is also the output of the nonlinear system of Eq. 5.1). Therefore, before the last hidden layer/output layer, we can add another hidden layer or multiple hidden layers with a sufficient number of neurons to approximate the nonlinear system of Eq. 5.1.*

5.2 Physics-based RNNs

In this section, we introduce three different methods to integrate domain knowledge into neural network modeling and training. The first method is to develop a hybrid model that integrates first-principles models with RNN models. The second method is to develop a partially-connected RNN structure using a priori knowledge of process input-output relationship. Lastly, a weight-constrained RNN model is developed by imposing constraints on the neural network weights based on the input-output relationship of the nonlinear system of Eq. 5.1.

5.2.1 Hybrid Model

While first-principles modeling has been studied and applied to chemical processes for over a century and has achieved good performances, it becomes difficult to obtain a 100% accurate first-principles model for large-scale systems due to inherent complexity. Therefore, in this chapter, we first propose a hybrid modeling method that introduces physical knowledge (e.g., first-principles knowledge based on physical laws such as mass and energy balances) into neural network modeling by combining a first-principles model and an RNN model together. Specifically, the hybrid model is developed using an RNN function $\tilde{f}_{nn}(x,u)$ to approximate the gap between the first-principles model and the actual nonlinear process as follows:

$$\dot{x} = \tilde{f}(x) + \tilde{g}(x)u + \tilde{f}_{nn}(x,u) \tag{5.4}$$

where $\dot{x} = \tilde{f}(x) + \tilde{g}(x)u$ is the first-principles model that is developed based on general physical laws and assumptions, and therefore, may not be able to fully capture the dynamics of the actual nonlinear processes of Eq. 5.1 due to mismatch between $\dot{x} = \tilde{f}(x) + \tilde{g}(x)u$ and $\dot{x} = f(x) + g(x)u$. The RNN function $\tilde{f}_{nn}(x,u)$ in Eq. 5.4 is utilized to bridge the gap between the first-principles knowledge and the real process data. It is demonstrated that the hybrid model of Eq. 5.4 has the following advantages compared with a fully-connected RNN model. First, the RNN in the hybrid model is only used to approximate the residual between first-principles models and real process

data, and therefore, may take less computing power and training time to learn. Additionally, when it comes to the operating region with no data available, the hybrid model can still be considered a reliable model due to its intrinsically physical knowledge, while the pure RNN model may be completely dysfunctional. For example, in [15], a hybrid model that combines first-principles free-falling equations and a neural network model was developed to improve the estimation of future trajectories of a paper ball being tossed, in which the neural network was developed to learn the model mismatch between the ground truth and the first-principles model-based solution. Additionally, in [192], a hybrid neural network model was developed for a chemical process where the linear part of the hybrid model is developed based on first-principles knowledge and the nonlinear term of reaction rate is provided by a neural network model using experiment/simulation data. It was demonstrated in [192] that the hybrid model achieved desired approximation performance and the neural network well approximated the nonlinear term of reaction rate that depends on multiple variables with an unknown reaction mechanism.

5.2.2 Partially-connected RNN

In industrial chemical processes, the unit operations in the upstream stage of the production process affect those in the downstream stage, while the impact is ignorable in the opposite direction. This connection between upstream and downstream stages is often reflected in the first-principles model (if there is any), and is barely incorporated in the development of a data-driven model for the entire process due to the difficulty of designing model structures. In particular, since it is not clear how to derive optimal architectures for process data without any a priori knowledge, fully-connected RNN networks are often state-of-the-art for large-scale, complex systems. Specifically, a black box NN model that takes all available inputs to predict the outputs of interest is preferred in developing a dynamic process model for the integrated upstream and downstream processes as it is easy to implement using open-source machine learning software and is able to account for all possible input-output relationships. As shown in Fig. 5.1, the RNN structure on the left represents a general RNN model (i.e., the fully-connected RNN model) with an input layer, a hidden layer consisting

of recurrent neurons, and an output layer, for which the training process follows the discussion of the three-step procedure in the previous section.

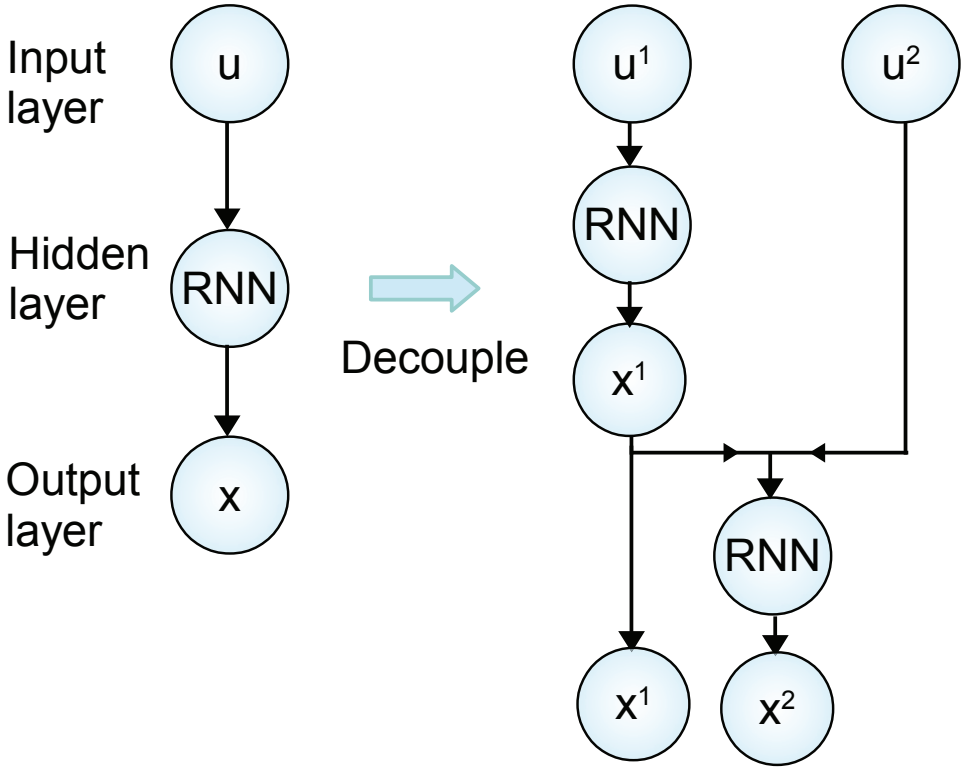


Figure 5.1: A partially-connected recurrent neural network structure based on process structural knowledge, where $u = [u^1,\ u^2]$ and $x = [x^1,\ x^2]$.

To account for the structural process knowledge into RNN modeling of the nonlinear system of Eq. 5.1, we develop a partially-connected RNN structure as shown on the right of Fig. 5.1. Specifically, we consider the nonlinear system of Eq. 5.1 under the assumption that the state vector x^1 is affected by u^1 only, and x^2 is affected by both u^1 and u^2, where $x = [x^1,\ x^2] \in \mathbf{R}^n$ and $u = [u^1 \in \mathbf{R}^{m_1},\ u^2 \in \mathbf{R}^{m_2}] \in \mathbf{R}^m$, $m_1 + m_2 = m$. It is shown in Fig. 5.1 that in the partially-connected RNN, u^1 only affects x^1, and both u^1 and u^2 have an impact on the output u^2. By partitioning the RNN modeling problem into two blocks (i.e., the data flows from u^1 to x^1, and from u^1, u^2 to x^1) that are corresponding to the structural a priori knowledge of the nonlinear system of Eq. 5.1, it is demonstrated that the hidden layers (i.e., the RNN layers) in the partially-connected RNN model are analogous to the unknown nonlinear functions of the system of Eq. 5.1 under the above assumption on input-output relationship. By explicitly changing the RNN structure to

exclude the connection between u^2 and x^1, a priori knowledge on process structure is infused into RNN modeling of the nonlinear system of Eq. 5.1, and therefore, an improved approximation performance can be derived. For example, due to the superiority of encoding priors into structure designs, in Remark 5.5, it is demonstrated that the number of hidden neurons and weight parameters could be significantly reduced to achieve the desired performance as good as the fully-connected model. Additionally, the partially-connected RNN model may need less training data to obtain a well-conditioned model since priors are acting to reveal the correct direction for RNN to converge to an optimal solution. Moreover, the partially-connected model may outperform the fully-connected model in the regime with no training data available since the model structure is consistent with the actual process state variable relationship of the nonlinear system of Eq. 5.1.

Remark 5.3. *It is noted that the partially-connected RNN model can achieve better approximation performance not only in regimes where data are not available, but also in the regime in which training and validation data are available. While in general the approximation performance of NN model on training dataset will be improved by increasing the number of neurons and parameters, excessive number of neurons may lead to over-fitting, which means that the NN model can capture the input-output relationship well for training dataset but not for the validation/testing datasets. Additionally, as in this particular example of Fig. 5.1, the second input u^2 does not affect the relationship between u^1 and x^1 (i.e., the weights between u^2 and x^1 should be zero), the connection between u^2 and x^1 in a fully-connected RNN model will instead result in a negative impact on the training process in terms of longer time to converge to an optimal solution.*

5.2.3 Weight-constrained RNN

Under the assumption that a portion of the input vector u^2 in the nonlinear system of Eq. 5.1 does not affect the output vector x^1, we develop an RNN model structure with constrained weight parameters representing the dynamic effects of process inputs u on the outputs x as shown in Fig. 5.2. Specifically, the weights connecting u^2 and x^1 (dashed gray lines in Fig. 5.2) are constrained in the RNN model such that the effects of u^2 on x^1 will be weakened during the

training process. Based on the RNN model of Eq. 5.3, the output vector x^1 and the hidden neuron r_i, $i = 1,...,h$ are derived as follows:

$$\dot{x}^1 = \sum_{i=1}^{h} w_i^{(2)} \dot{r}_i \tag{5.5}$$

$$\dot{r}_i = -a_i r_i + \theta_i y \tag{5.6}$$

where $\theta_i = b_i[w_{1i}^{(1)},...,w_{hi}^{(1)},...,w_{(h+m)i}^{(1)}]$ and $y = [\sigma(r_1),...,\sigma(r_h),u^1,u^2]^T$. a_i,b_i are constants, $w_{ji}^{(1)}$ is the weight connecting the jth input, $j = 1,...,h+m$ to ith neuron, $i = 1,...,h$, and y is the input vector consisting of the hidden states r and the manipulated inputs u. $w^{(1)}$, $w^{(2)}$ represent the weight vectors before and after the hidden layer. Similarly, the bias term is not included in the notation since it can be considered as an additional constant input. Based on Eq. 5.5 and Eq. 5.6, the following equation is derived to demonstrate the contribution of u^2 to $\dot{x}^1$:

$$\begin{aligned} \dot{x}^1 &= \sum_{i=1}^{h} w_i^{(2)}(-a_i r_i + \theta_i y) \\ &= \sum_{i=1}^{h} n_i(r,w) + w_i^{(2)} b_i([w_{(h+1)i}^{(1)},...,w_{(h+m_1)i}^{(1)}]u^1 + [w_{(h+m_1+1)i}^{(1)},...,w_{(h+m)i}^{(1)}]u^2) \end{aligned} \tag{5.7}$$

where $n_i(\cdot,\cdot)$ is a nonlinear function of the neuron states r and weights w. Therefore, to reduce the impact of u^2 to $\dot{x}^1$, the weight product $\Pi_w = |w_i^{(2)} b_i[w_{(h+m_1+1)i}^{(1)},...,w_{(h+m)i}^{(1)}]|$ should be constrained by a sufficiently small bound, and this constraint will also be incorporated in the training of the RNN model with the above input-output relationship. It is noted that since the constraint is applied on the weight product Π_w for the weight-constrained RNN model in Fig. 5.2, a zero bound for the weight constraint could lead to disconnection of hidden neurons from inputs and outputs, and therefore, should be avoided in any case. In addition to the weight constraints, penalty components on weight parameters can be employed in the loss function of the RNN optimization problem to introduce a priori weight knowledge into the training process. In general, regularization techniques (e.g., L1 and L2 regularization) are utilized in the training process of a neural network model to obtain a less complex model and avoid over-fitting in the presence of a large number of features in datasets. Therefore, to constrain the weight product Π_w in Eq. 5.7, the following loss function is

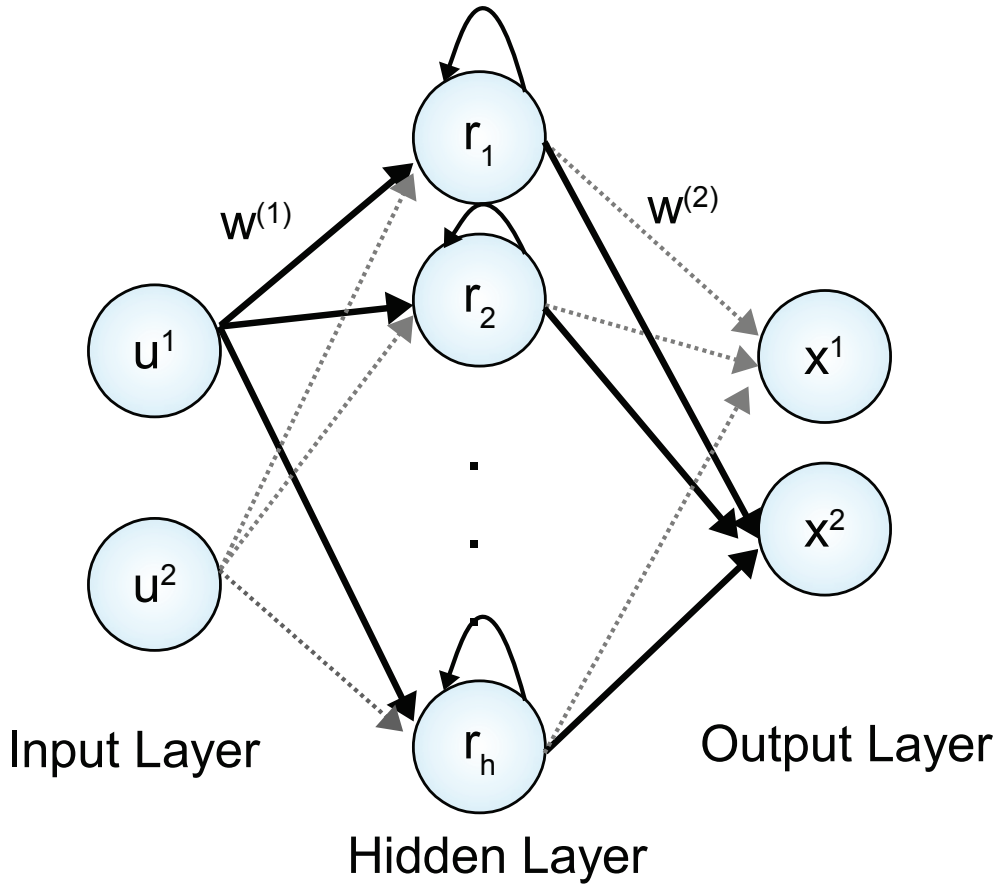


Figure 5.2: A weight-constrained recurrent neural network structure, where $w^{(1)}$ and $w^{(2)}$ are the weights before and after the hidden layer, r_i, $i = 1, ..., h$ is the RNN hidden neuron, and the dashed gray lines denote the diminished connections between u^2 and x^1.

developed:

$$L = \sum_{i=1}^{N_d} (x_i - \hat{x}_i)^2 + \lambda \Pi_w \tag{5.8}$$

where x_i and $\hat{x}_i$ are the actual and predicted outputs, respectively, N_d is the number of data samples in the dataset, and $\lambda > 0$ is the weight for the regularization term. It is noted that λ needs to be carefully chosen such that the regularization term can effectively decrease the values of the weights, but does not dominate the optimization problem of training a neural network model. Specifically, while a nonzero λ is required to penalize the regularization term, a large λ may render the optimization problem under-fitting due to the dominance of the regularization term in the loss function of Eq. 5.8. Therefore, we evaluate λ against any metric (e.g., mean-squared error, among many other criteria) and select the value of λ that achieves the desired approximation performance

on training and validation datasets.

Alternatively, to fully remove the connection between u^2 and x^1, we can design another set of neurons $r_{h+1},...,\ r_{2h}$ in the hidden layer as shown in Fig. 5.3. It is demonstrated that u^2 is disconnected from the neurons $r_1,...,\ r_h$ that contribute to the output vector x^1 to eliminate the impact of u^2 on x^1. As a result, to maintain the impact of inputs on the other output vector x^2, the new set of neurons $r_{h+1},...,\ r_{2h}$ are utilized in the hidden layer to connect both the inputs u^1 and u^2 to the output x^2. It is noted that compared to a fully-connected RNN model, the number of neurons and the number of weights in the weight-constrained RNN shown in Fig. 5.3 are increased to separate the connections to multiple output vectors.

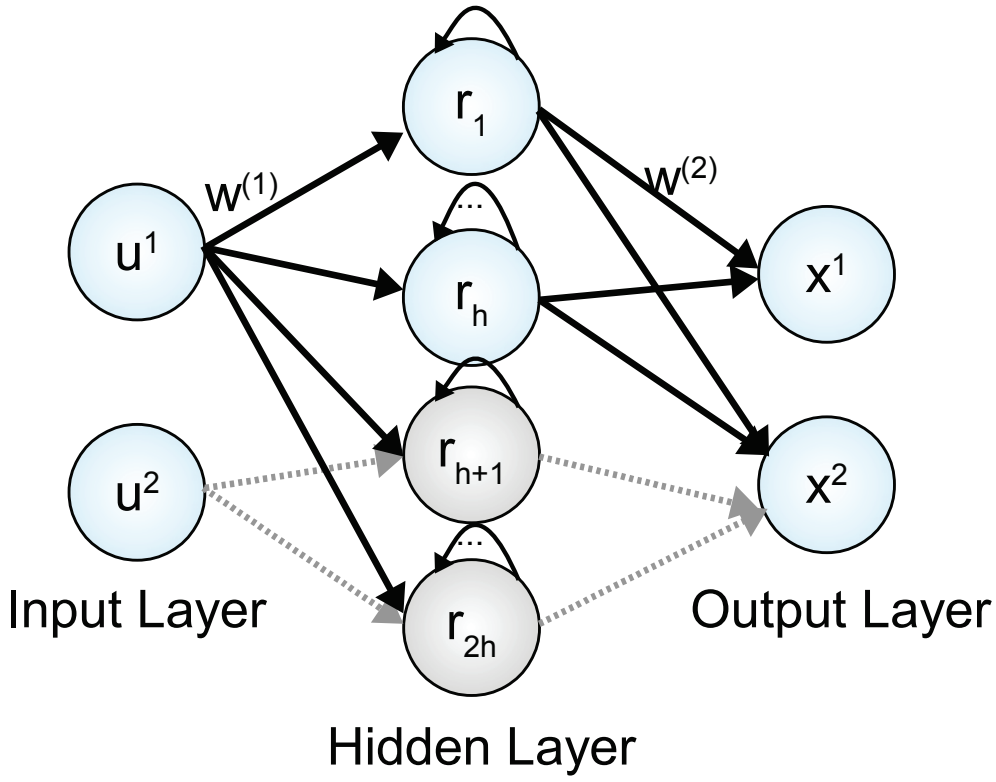


Figure 5.3: A recurrent neural network structure, where the connection between u^2 and x^1 is fully removed from the blue neurons, and the connection between u^2 and x^2 is rebuilt using the gray neurons in the hidden layer.

Based on the RNN model of Eq. 5.3, the output vector x and the hidden neuron r_i, $i = 1,...,2h$

in Fig. 5.3 are derived as follows:

$$\dot{x}^1 = \sum_{i=1}^{h} w_i^{(2)} \dot{r}_i, \quad \dot{x}^2 = \sum_{i=1}^{2h} w_i^{(2)} \dot{r}_i \tag{5.9}$$

$$\dot{r}_i = -a_i r_i + \theta_i y, \; i = 1, ..., 2h \tag{5.10}$$

where $\theta_i = b_i[w_{1i}^{(1)}, ..., w_{(2h)i}^{(1)}, ..., w_{(2h+m)i}^{(1)}]$and $y = [\sigma(r_1), ..., \; \sigma(r_{2h}), u^1, u^2]^T$. a_i and b_i are constants, $w_{ji}^{(1)}$ is the weight connecting the jth input, $j = 1, ..., 2h+m$ to ith neuron, $i = 1, ..., 2h$, and y is the input vector consisting of the hidden states r and the manipulated inputs u. $w^{(1)}$, $w^{(2)}$ represent the weight vectors before and after the hidden layer. Specifically, to train the weight-constrained RNN model with the structure of Fig. 5.3, we first develop a fully-connected RNN model and then let the weights between u^2 and r_i, $i = 1, ..., h$, and the weights between r_i, $i = h+1, ..., 2h$ and x^1 (denoted by $\tilde{w}$) be zero or be constrained by a sufficiently small bound. Unlike the weight-constrained RNN model in Fig. 5.2, the weight constraint for the RNN model with the structure of Fig. 5.3 can be equipped with a zero bound such that the connections can be fully removed for the network. Additionally, the above weight constraints on the RNN weights need to be well-defined before training. It should be noted that since there exist three types of weight matrices in an RNN model: 1) the weight matrix connecting the input layer and the hidden layer, 2) the weight matrix feeding the past neuron information into the current network (i.e., the feedback loop in r_i, $i = 1, ..., 2h$), and 3) the weight matrix connecting the hidden layer to the output layer, the constraints need to be implemented in all the three weight matrices such that u^2 and x^1 are fully disconnected.

We train the above weight-constrained RNN model in Keras, and implement weight constraints in the *constraints.py* source file. Specifically, the constraints on the weight matrices that connect inputs to hidden neurons and hidden neurons to outputs are activated through the argument `kernal_constraint`. The weight matrix feeding the past neuron information to the current network is implemented by invoking `recurrent_constraint`. Additionally, to develop an RNN model that obtains the optimal weights subject to the weight constraints, the RNN optimizer

(e.g., *adaptive learning rate optimization algorithm*) needs to be modified to minimize the loss function while accounting for the weight constraints in the optimization problem. Alternatively, the weight constraints can be implemented at the end of each training epoch such that the weights that meet the constraints remain unchanged and those exceeding the constraints will be bounded to the saturation value. The saturated weights will then be utilized as the initial condition for the optimization problem for the next training epoch, and the above process is repeated until the stopping criteria of the training process are satisfied.

Remark 5.4. *In this section, we proposed two approaches for weight-constrained RNN models. Specifically, the first approach (i.e., Fig. 5.2) is to develop a weight-constrained RNN model by adding a regularization term on constrained weights in the training process of RNN models to reduce the connection between u^2 and x^1. This is typically used for the systems where we know a priori that the connections between certain inputs and outputs are weakly connected (but not fully unconnected). However, the second approach (i.e., Fig. 5.3) is to develop a weight-constrained RNN model by adding another set of neurons such that the connections between u^2 and x^1 are fully removed. Therefore, it will be applied to the systems where some of the inputs do not affect the outputs at all.*

5.2.4 RNN Training Process

All the physics-based RNN models are developed using Keras library, an open-source neural-network library written in Python. Specifically, the hybrid model is developed following the construction method for a fully-connected RNN model, where the training dataset is preprocessed to represent the gap between the first-principles model and real process data, and then separated into training, validation and testing datasets. To develop a partially-connected RNN model in Fig. 5.1, an RNN layer is first developed to connect u^1 and x^1. Subsequently, x^1 and u^2 are concatenated and followed by a second RNN layer to ultimately obtain x^2. It is noted that instead of using the full input and output vectors u and x, the input vectors u^1, u^2 and the output vectors x^1, x^2 need to be specified and fed into the partially-connected RNN model separately. Therefore, the

inputs and the outputs to the partially-connected RNN model in the training process are defined as $[u^1,\, u^2]$ and $[x^1,\, x^2]$, respectively. The development of a weight-constrained RNN model in Fig. 5.2 follows that for a fully-connected RNN model except that the weight constraints need to be added in the optimization problem of RNN training process beforehand by updating Keras optimizer source files. However, to develop a weight-constrained RNN model with the structure of Fig. 5.3, we can either implement the weight constraints within each epoch of the optimization process in Keras optimizer source files, or saturate the corresponding elements in the weight matrices at the end of each epoch to update the initial guess of weights for the next training epoch in Keras constraints source files. The training processes for both weight-constrained RNN models follow that for a fully-connected RNN model, where the training and validation datasets are used to obtain the optimal weight matrices for RNN models, and the testing dataset is used to evaluate their prediction performances.

To prevent the weights from drifting to infinity during the RNN training process, the weight vector θ_i of the RNN model of Eq. 5.3 is also bounded by $|\theta_i| \leq \theta_m$, with $\theta_m > 0$. It is noted that while it is possible to obtain a theoretical value for θ_m, this value will usually be conservative. Therefore, in the implementation of the RNN training process, we give θ_m a reasonable value and see if the RNN can approximate the nonlinear system of Eq. 5.1 with the satisfaction of the modeling error constraint. The interested reader may refer to [80], where a σ-modification is utilized in the RNN learing algorithm to ensure that the weights are bounded during the training process. Additionally, the hybrid model, the partially-connected RNN model, and the weight-constrained RNN model are all trained with a constraint on the modeling error, i.e., $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma|x|$, where $\gamma > 0$, such that the obtained RNN models can well represent the actual nonlinear process of Eq. 5.1 and can be utilized in a model-based predictive controller that stabilizes the system at its steady-state with guaranteed stability. The detailed RNN learning algorithm and the proof of the boundedness of RNN modeling error can be found in [181].

Remark 5.5. *Consider the nonlinear system of Eq. 5.1 with $x = [x^1,\, x^2] \in \mathbf{R}^n$ and $u = [u^1,\, u^2] \in \mathbf{R}^n$, where x^1 and x^2, u^1 and u^2 are of the same dimension, respectively (i.e., $x^1, x^2 \in \mathbf{R}^{\frac{n}{2}}$, $u^1, u^2 \in \mathbf{R}^{\frac{m}{2}}$).*

Under the assumption of the input-output relationship in this section, the total number of weights for a partially-connected RNN model with two hidden layers, where each hidden layer has h neurons, is calculated to be $\frac{3}{2}nh+mh+2h^2$*, while the total number of weights for a fully-connected RNN model with the same two hidden layers is* $mh+3h^2+nh$ *(the bias term is ignored in the comparison as it can be considered a constant input node). Since in most cases, the number of neurons is much greater than the number of inputs and states to achieve a desired approximation performance, the number of weights for a decoupled RNN model is significantly reduced due to the incorporation of process structural knowledge (*$\frac{3}{2}nh+mh+2h^2 << nh+mh+3h^2$ *when* $h >> m,n$*). However, it is noted that the number of weights in a weight-constrained model with the structure of Fig. 5.3 is increased compared to the fully-connected RNN model due to the new set of hidden neurons that are used to rebuild the connection between* u^2 *and* x^2*.*

Remark 5.6. *In addition to the weight constraints as discussed above, regularization can be utilized to penalize the weights that need to be constrained in the loss function of Eq. 5.8. The implementation of regularization using Keras is as follows. First, the* `kernel_regularizer` *command is invoked to activate the regularization term in the loss function in each layer with the desired regularization technique (i.e., L1 or L2 regularization). With this, all the elements in the weight matrices that connect the inputs to the hidden neurons and connect the hidden neurons to the outputs are penalized in the loss function with a regularization parameter* λ *as introduced in Eq. 5.8. Subsequently, to ensure that only the weights that need to be constrained are penalized in the loss function, the regularizers.py source file is adapted in which the corresponding weights for the undesired connections are included. Finally, the weight matrix feeding the past neuron information into the current network is penalized following the same strategy as discussed above for the implementation of weight constraints.*

Remark 5.7. *It is noted that all the RNN models in this section are developed for the nominal system of Eq. 5.1 without disturbances. However, in the presence of time-varying disturbances, the RNN model that is trained for the nominal system may be dysfunctional in a model-based predictive controller due to a considerable model mismatch. To that end, online update of RNN models can be*

employed to capture the nonlinear dynamics subject to disturbances using the most recent process measurement data. The interested readers may refer to [178] for the details of implementation of online RNN update.

Remark 5.8. *In the case that a single RNN model is not able to well represent the dynamics of the nonlinear process of Eq. 5.1 in the entire operating region, multiple RNN models can be developed to improve the overall prediction accuracy in the context of ensemble learning [96,189]. The development of multiple RNN models via ensemble learning can be found in [182], where k different RNN models were developed for the same nonlinear process based on a k-fold cross-validation and were utilized to derive a final prediction result that was significantly improved compared to a single RNN model. Additionally, in [182], to improve computational efficiency of ensemble learning and multiple RNN predictions in the real-time implementation of machine-learning-based predictive controllers, parallel computing can be employed to speed up the computation of RNN predictions using multiple compute cores in a distributed computing cluster. It was also demonstrated in [182] that the computation time of calculating multiple RNN prediction results in an RNN-based predictive controller was significantly reduced under parallel computation of the ensemble of RNN models.*

Remark 5.9. *To extend the proposed RNN modeling methods to high-dimensional systems, it is necessary to find the relationships between the inputs and outputs. For example, we can introduce sparsity regularization that is similar to Eq. 5.8 but with relatively large penalty, or we can also apply well-established methods such as relative gain array to determine the best input-output pairings for multivariable processes.*

5.3 RNN-based Predictive Control

In this section, we incorporate the RNN model developed in the previous section into the design of model-based predictive controllers to optimize process performance while guaranteeing closed-loop stability. Specifically, for any initial condition $x_0 \in \Omega_\rho$, a Lyapunov-based model

predictive controller (LMPC) is developed to drive the closed-loop state of the nonlinear system of Eq. 5.1 to the steady-state while maintaining the state in the stability region Ω_ρ for all times. Subsequently, Lyapunov-based economic model predictive controller (LEMPC) is developed using the RNN model to optimize process economic performance with guaranteed boundedness of the state in Ω_ρ for all times.

5.3.1 Lyapunov-based MPC using RNN models

The Lyapunov-based model predictive control (LMPC) using the RNN model of Eq. 5.3 is utilized to stabilize the nonlinear system of Eq. 5.1 in the stability region. The formulation of LMPC optimization problem is given as follows:

$$\mathscr{J} = \min_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} (\tilde{x}^T Q\tilde{x} + u^T Ru)dt \tag{5.11a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), u(t)) \tag{5.11b}$$

$$u(t) \in U,\ \forall\ t \in [t_k, t_{k+N}) \tag{5.11c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{5.11d}$$

$$\dot{V}(x(t_k), u) \leq \dot{V}(x(t_k), \Phi_{nn}(x(t_k))),$$
$$\text{if } x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{nn}} \tag{5.11e}$$

$$V(\tilde{x}(t)) \leq \rho_{nn},\ \forall\ t \in [t_k, t_{k+N}),\ \text{if } x(t_k) \in \Omega_{\rho_{nn}} \tag{5.11f}$$

where $\tilde{x}$ is the predicted state trajectory, $S(\Delta)$ is the set of piecewise constant functions with period Δ, N is the number of sampling periods in the prediction horizon, and $\dot{V}(x,u)$represents $\frac{\partial V(x)}{\partial x}(F_{nn}(x,u))$. In the optimization problem of Eq. 5.11, the objective function of Eq. 5.11a is the integral of the cost function $l(\tilde{x},t) = (\tilde{x}^T Q\tilde{x} + u^T Ru)$ over the prediction horizon, where $l(0,0) = 0$ and $l(\tilde{x},t) > 0$, $\forall (\tilde{x},t) \neq (0,0)$. The constraint of Eq. 5.11b is the RNN model of Eq. 5.3 that is used to predict the states of the closed-loop system. Eq. 5.11c defines the input constraints applied over the entire prediction horizon. Eq. 5.11d defines the initial condition $\tilde{x}(t_k)$ of Eq. 5.11b, which

is the state measurement at $t = t_k$. The constraint of Eq. 5.11e forces the closed-loop state to move towards the origin if $x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{nn}}$. However, if $x(t_k)$ enters $\Omega_{\rho_{nn}}$, the states predicted by the RNN model of Eq. 5.11b will be maintained in $\Omega_{\rho_{nn}}$ for the entire prediction horizon. The LMPC of Eq. 5.11 is implemented in a sample-and-hold fashion, i.e., an optimal input trajectory $u^*(t)$, $t \in [t_k, t_{k+N})$ is obtained by solving the LMPC optimization problem of Eq. 5.11 at each sampling time, from which only the control action for the first sampling period of the prediction horizon will be applied. In [181], it is demonstrated that under the LMPC of Eq. 5.11, the state of the nonlinear system of Eq. 5.1 is bounded in the stability region Ω_ρ for all times, and can ultimately converge to the origin provided that the modeling error between the the nonlinear system of Eq. 5.1 and the RNN model of Eq. 5.3 is sufficiently small. Detailed proof for closed-loop stability can be found in [181] and is omitted here due to space limitations.

5.3.2 Lyapunov-based EMPC using RNN models

The Lyapunov-based economic model predictive control (LEMPC) using the RNN model of Eq. 5.3 is utilized to optimize process economic performance while maintaining the closed-loop state of the nonlinear system of Eq. 5.1 in the stability region Ω_ρ. The LEMPC is formulated by the following optimization problem:

$$\mathscr{J} = \max_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(t), u(t)) dt \tag{5.12a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), u(t)) \tag{5.12b}$$

$$u(t) \in U, \ \forall \ t \in [t_k, t_{k+N}) \tag{5.12c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{5.12d}$$

$$V(\tilde{x}(t)) \leq \rho_e, \ \forall \ t \in [t_k, t_{k+N}), \text{ if } x(t_k) \in \Omega_{\rho_e} \tag{5.12e}$$

$$\dot{V}(x(t_k), u) \leq \dot{V}(x(t_k), \Phi_{nn}(x(t_k))), \quad \text{if } x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_e} \tag{5.12f}$$

where the notations follow those in Eq. 5.11. The optimization problem of Eq. 5.12 maximizes the objective function of Eq. 5.12a that integrates $l_e(\tilde{x}(t), u(t))$ over the prediction horizon subject to the constraints of Eqs. 5.12b-5.12f. Specifically, the constraint of Eqs. 5.12b-5.12d are the same as Eqs. 5.11b-5.11d for LMPC. The constraint of Eq. 5.12e maintains the predicted closed-loop states in Ω_{ρ_e} if $x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_e}$, where Ω_{ρ_e}, $0 < \rho_e < \rho$, is a level set of Lyapunov function that guarantees the boundedness of state in the closed-loop stability region Ω_ρ accounting for the model mismatch between the RNN model of Eq. 5.12b and the nonlinear process of Eq. 5.1. On the other hand, if $x(t_k)$ leaves Ω_{ρ_e}, the contractive constraint of Eq. 5.12f will be activated to drive the state towards the origin within the next sampling period. It is demonstrated that the closed-loop state of the nonlinear system of Eq. 5.1 is bounded in the stability region Ω_ρ for all times under the LEMPC of Eq. 5.12. The detailed proof of closed-loop stability under LEMPC is given in [181].

Remark 5.10. *It is demonstrated in [181] that closed-loop stability is guaranteed for the nonlinear system of Eq. 5.1 under the RNN-based MPC of Eq. 5.11 provided that the modeling error between the RNN model and the actual nonlinear system is sufficiently small. Specifically, the constraints of Eq. 5.11e and Eq. 5.11f are developed to guarantee that the closed-loop state will move towards the origin and can be ultimately bounded in a small neighborhood around the origin regardless of the length of prediction horizon. However, it is noted that the use of a longer prediction horizon in MPC can generally improve closed-loop performance by obtaining better solutions that lead to less control energy consumption and smoother state trajectories. Though we did not show the detailed proof for closed-loop stability due to space limitation in this chapter, the proposed partially-connected RNN and weight-constrained RNN models that account for process structural knowledge by modifying the structure of RNNs, and adding certain constraints in the training process of RNNs, respectively, are developed satisfying the modeling error constraint. Therefore, closed-loop stability can be established for the closed-loop MPC using the proposed RNN models.*

5.4 Application to a Chemical Process Example

A chemical process example is utilized to demonstrate the application of the proposed RNN modeling with the incorporation of structural process knowledge. Specifically, two well-mixed, non-isothermal continuous stirred tank reactors (CSTR) in series are considered where an irreversible second-order exothermic reaction takes place in each reactor as shown in Fig. 5.4. The reaction transforms a reactant A to a product B ($A \rightarrow B$). Each of the two reactors are fed with reactant material A with the inlet concentration C_{Aj0}, the inlet temperature T_{j0} and feed volumetric flow rate of the reactor F_{j0}, $j = 1,2$, where $j = 1$ denotes the first CSTR and $j = 2$ denotes the second CSTR. Each CSTR is equipped with a heating jacket that supplies/removes heat at a rate Q_j, $j = 1,2$. The CSTR dynamic models are described by the following material and energy balance equations:

$$\frac{dC_{A1}}{dt} = \frac{F_{10}}{V_1}(C_{A10} - C_{A1}) - k_0 e^{\frac{-E}{RT_1}} C_{A1}^2 \tag{5.13a}$$

$$\frac{dT_1}{dt} = \frac{F_{10}}{V_1}(T_{10} - T_1) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT_1}} C_{A1}^2 + \frac{Q_1}{\rho_L C_p V_1} \tag{5.13b}$$

$$\frac{dC_{B1}}{dt} = -\frac{F_{10}}{V_1} C_{B1} + k_0 e^{\frac{-E}{RT_1}} C_{A_1}^2 \tag{5.13c}$$

$$\frac{dC_{A2}}{dt} = \frac{F_{20}}{V_2} C_{A20} + \frac{F_{10}}{V_2} C_{A1} - \frac{F_{10} + F_{20}}{V_2} C_{A2} - k_0 e^{\frac{-E}{RT_2}} C_{A2}^2 \tag{5.13d}$$

$$\frac{dT_2}{dt} = \frac{F_{20}}{V_2} T_{20} + \frac{F_{10}}{V_2} T_1 - \frac{F_{10} + F_{20}}{V_2} T_2 + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT_2}} C_{A2}^2 + \frac{Q_2}{\rho_L C_p V_2} \tag{5.13e}$$

$$\frac{dC_{B2}}{dt} = \frac{F_{10}}{V_2} C_{B1} - \frac{F_{10} + F_{20}}{V_2} C_{B2} + k_0 e^{\frac{-E}{RT_2}} C_{A2}^2 \tag{5.13f}$$

where C_{Aj}, V_j, T_j and Q_j, $j = 1,2$ are the concentration of reactant A, the volume of the reacting liquid, the temperature, and the heat input rate in the first and the second reactor, respectively. The reacting liquid has a constant density of ρ_L and a heat capacity of C_p for both reactors. ΔH, k_0, E, and R represent the enthalpy of the reaction, pre-exponential constant, activation energy, and ideal gas constant, respectively, and are the same for both reactors. Process parameter values are listed in Table 5.1.

Table 5.1: Parameter values of the two CSTRs in series.

$T_{10} = 300\ K$	$T_{20} = 300\ K$
$F_{10} = 5\ m^3/hr$	$F_{20} = 5\ m^3/hr$
$V_1 = 1\ m^3$	$V_2 = 1\ m^3$
$T_{1_s} = 402\ K$	$T_{2_s} = 402\ K$
$C_{A1_s} = 1.95\ kmol/m^3$	$C_{A2_s} = 1.95\ kmol/m^3$
$C_{A10_s} = 4\ kmol/m^3$	$C_{A20_s} = 4\ kmol/m^3$
$Q_{1_s} = 0.0\ kJ/hr$	$Q_{2_s} = 0.0\ kJ/hr$
$k_0 = 8.46 \times 10^6\ m^3/kmol\ hr$	$\Delta H = -1.15 \times 10^4\ kJ/kmol$
$C_p = 0.231\ kJ/kg\ K$	$R = 8.314\ kJ/kmol\ K$
$\rho_L = 1000\ kg/m^3$	$E = 5 \times 10^4\ kJ/kmol$

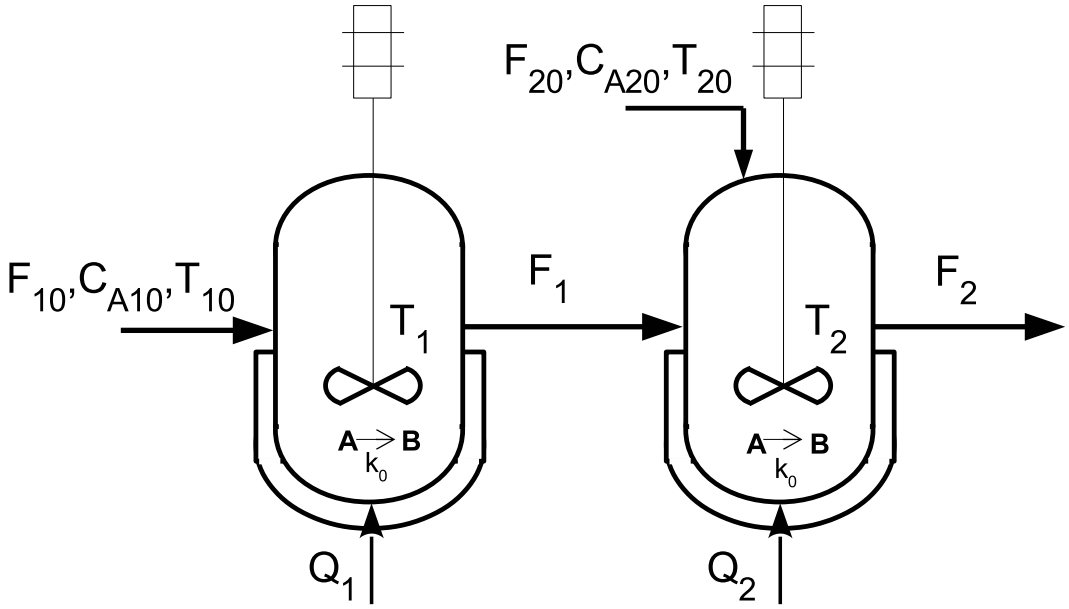


Figure 5.4: Process flow diagram of two CSTRs in series.

The manipulated inputs for both CSTRs are the inlet concentration of species A and the heat input rate, which are represented by the deviation variables $\Delta C_{Aj0} = C_{Aj0} - C_{Aj0_s}$, $\Delta Q_j = Q_j - Q_{j_s}$, $j = 1,2$, respectively. The manipulated inputs are bounded as follows: $|\Delta C_{Aj0}| \leq 3.5\ kmol/m^3$ and $|\Delta Q_j| \leq 5 \times 10^5\ kJ/hr$, $j = 1,2$. Therefore, the states and the inputs of the closed-loop system are $x^T = [C_{A1} - C_{A1_s}\ \ T_1 - T_{1_s}\ \ C_{A2} - C_{A2_s}\ \ T_2 - T_{2_s}]$ and $u^T = [\Delta C_{A10}\ \ \Delta Q_1\ \ \Delta C_{A20}\ \ \Delta Q_2]$, respectively,

where C_{A1_s}, C_{A2_s}, T_{1_s} and T_{2_s} are the steady-state values of concentration of A and temperature in the first and second reactors, such that the equilibrium point of the system is at the origin of the state-space.

The explicit Euler method with an integration time step of $h_c = 10^{-4}$ hr is used to numerically simulate the dynamic model of Eq. 5.13. The nonlinear optimization problems of the LMPC of Eq. 5.11 and of the LEMPC of Eq. 5.12 are solved using the python module of the IPOPT software package [158], named PyIpopt with the sampling period $\Delta = 10^{-2}$ hr. Two control Lyapunov functions $V_1(x) = x^T P_1 x$, and $V_2(x) = x^T P_2 x$ are designed for two CSTRs, respectively, with the following positive definite P matrices:

$$P_1 = P_2 = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{5.14}$$

The closed-loop stability regions for the two CSTRs are characterized with $\rho = 380$, where $\rho_e = 260$ is chosen for the LEMPC of Eq. 5.12.

Open-loop Simulation on Testing Dataset

Open-loop simulations are first carried out to demonstrate the open-loop prediction performances of the fully-connected RNN model, the partially-connected RNN model, and the weight-constrained RNN model, respectively. The development of an RNN model for the CSTR process of Eq. 5.13 follows that in [182]. It should be noted that all the RNN models are developed using the same dataset with the same neural network parameters as follows: 2 hidden layers with 30 neurons in each layer, *tanh* as the activation function, and *Adam* as the optimizer. The root mean square errors (RMSE) between the first-principles state trajectories (i.e., the state trajectories using the first-principles model of Eq. 5.13) and the above three models, respectively, are reported in Table 5.2, where P-RNN, W-RNN and F-RNN represent the partially-connected RNN model, the weight-constrained RNN model, and the fully-connected RNN model, respectively.

From Table 5.2, it is demonstrated that the partially-connected RNN model and the

weight-constrained RNN model outperform the fully-connected model in that the open-loop approximations of C_{A1}, C_{A2}, T_1 and T_2 are significantly improved.

Table 5.2: RMSE comparison of open-loop prediction results with the first-principles model results.

	P-RNN	W-RNN	F-RNN
C_{A1} $(kmol/m^3)$	1.0×10^{-4}	5.6×10^{-6}	0.9×10^{-4}
T_1 (K)	0.14	0.018	0.15
C_{A2} $(kmol/m^3)$	8.2×10^{-7}	2.0×10^{-6}	2.6×10^{-6}
T_2 (K)	5.4×10^{-4}	0.0076	0.049

Remark 5.11. *It is noted that the comparison results in Table 5.2 were generated using extensive open-loop simulations with various initial conditions and control actions, under which the superiority of the proposed modeling approaches is clearly demonstrated by showing that the partially-connected RNN model and the weight-constrained RNN model outperform the fully-connected RNN model in terms of better approximation performance in the entire operating region.*

Closed-loop Simulation under LMPC

After demonstrating the open-loop prediction performances of the fully-connected RNN, the partially-connected RNN and the weight-constrained RNN for the CSTR process of Eq. 5.13 in the stability region, we perform the closed-loop simulation under the LMPC of Eq. 5.11 using the above three models, respectively. Additionally, the closed-loop simulation under the LMPC of Eq. 5.11 using the first-principles model of Eq. 5.13 is added as a baseline for comparison. The control objective of RNN-based LMPC is to operate the CSTR process of Eq. 5.13 at its steady-state while maintaining the closed-loop state in the stability region Ω_ρ for all times.

In Fig. 5.5, it is demonstrated that all the states (i.e., C_{A1}, T_1, C_{A2} and T_2) converge to the origin within 0.05 *hr* under the LMPC using the partially-connected RNN model and the

weight-constrained RNN model. However, under the LMPC using a fully-connected RNN model, the concentration in the first CSTR (i.e., C_{A1}) shows undesirable oscillations around the origin due to its considerable model mismatch as reported in Table 5.2. Therefore, through open-loop and closed-loop simulations, the partially-connected RNN model and the weight-constrained RNN model that incorporate structural process knowledge of the CSTR process of Eq. 5.13 are demonstrated to achieve better approximation performance than the fully-connected RNN model.

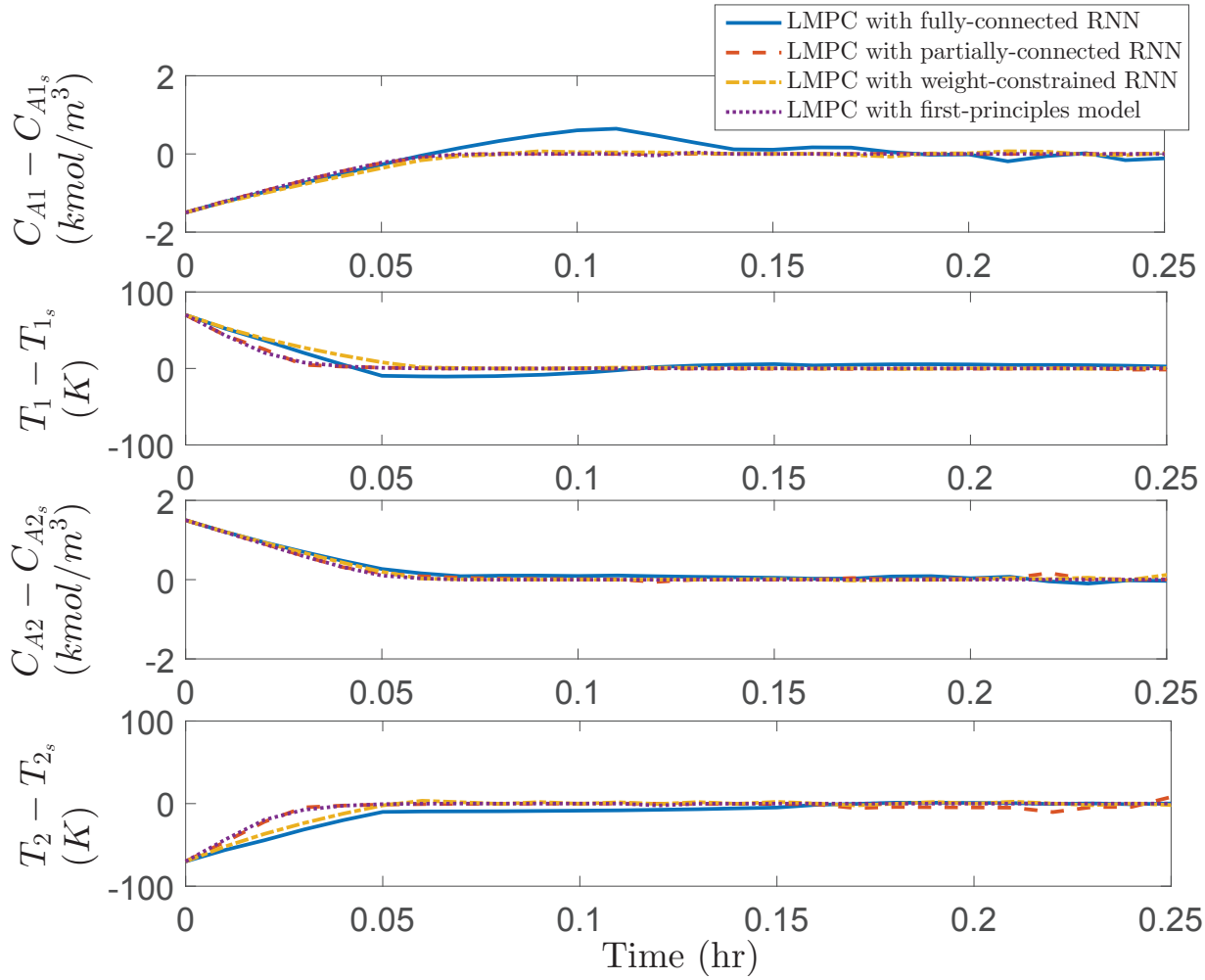


Figure 5.5: The state profiles ($C_{A1} - C_{A1_s}$, $T_1 - T_{1_s}$, $C_{A2} - C_{A2_s}$ and $T_2 - T_{2_s}$) for the closed-loop simulation of two CSTRs in series under the LMPC using the fully-connected RNN, the partially-connected RNN, the weight-constrained RNN, and the first-principles model of Eq. 5.13, respectively, for an initial condition (-1.5, 70, 1.5, -70).

Closed-loop Simulation under LEMPC

The control objective of LEMPC is to maximize the profit of both CSTR systems described in Eq. 5.13 by manipulating the inlet concentration ΔC_{A10} and C_{A20} and the heat inputs rate ΔQ_1 and ΔQ_2, and meanwhile maintain the closed-loop state trajectories in the stability region Ω_ρ for all times under LEMPC. The objective function of the LEMPC optimizes the production rate of B as follows:

$$l_e(\tilde{x},u) = k_0 e^{-E/RT_1} C_{A1}^2 + k_0 e^{-E/RT_2} C_{A2}^2 \tag{5.15}$$

Closed-loop simulations are performed under the LEMPC of Eq. 5.12 using the first-principles model of Eq. 5.13 and the three RNN models, respectively. In Fig. 5.6, it is demonstrated that the state trajectories for both CSTRs are bounded in the stability region Ω_ρ for all times under LEMPC. Fig. 5.7 shows the evolution the Lyapunov function values of V_1 and V_2 under LEMPC using the first-principles model of Eq. 5.13 and three different RNN models, respectively. Specifically, due to a relatively large model mismatch for the fully-connected RNN model as reported in Table 5.2, the contractive constraint of Eq. 5.12f is activated frequently under the LEMPC using a fully-connected RNN model because the actual process state does not stay in Ω_{ρ_e} under the constraint of Eq. 5.12e. As a result, it is observed in Fig. 5.7 that the V profiles under the fully-connected model show larger oscillation compared to those under the other two RNN models and under the first-principles model.

Additionally, we compare the accumulated economic profits $L_E = \int_0^{t_p} L_e(x,u)dt$ within the operation period $t_p = 0.32\ hr$ for the closed-loop CSTRs under the steady-state operation (i.e., the CSTRs are operated at their steady-states for all times), and the LEMPC using the first-principles model of Eq. 5.13 and the three RNN models, respectively. The result is shown in Fig. 5.8, from which it is demonstrated that the closed-loop operation under LEMPC achieves higher economic profits than the steady-state operation. Specifically, the LEMPC using the first-principles model achieves the highest economic benefits since the closed-loop state trajectory reaches and stays at the boundary of Ω_{ρ_e} smoothly based on accurate predictions. Moreover, it is demonstrated that

the LEMPC using the partially-connected RNN model and the weight-constrained RNN model economically outperform that under the fully-connected RNN model due to better prediction accuracy in the stability region. Therefore, through both open-loop and closed-loop simulations, we demonstrate that the physics-based RNN models achieve desired approximation performance for the CSTR process of Eq. 5.13 and provide reliable state predictions for model-based predictive controllers.

5.5 Conclusion

In this chapter, we developed three modeling approaches that incorporates a priori process knowledge into RNN models. Specifically, a hybrid model that combines a first-principles model and an RNN model was first developed. Then, a partially-connected RNN model and a weight-constrained RNN model were developed based on an assumption on process input-output relationship. The partially-connected and the weight-constrained RNN models were then utilized in RNN-MPC and RNN-EMPC, and applied to a chemical process example, from which it was demonstrated that the open-loop and closed-loop prediction performances under the LMPC and the LEMPC using the above two RNN models outperformed those under the LMPC and LEMPC using a fully-connected RNN model in terms of higher prediction accuracy, smoother state trajectories, and better economic benefits.

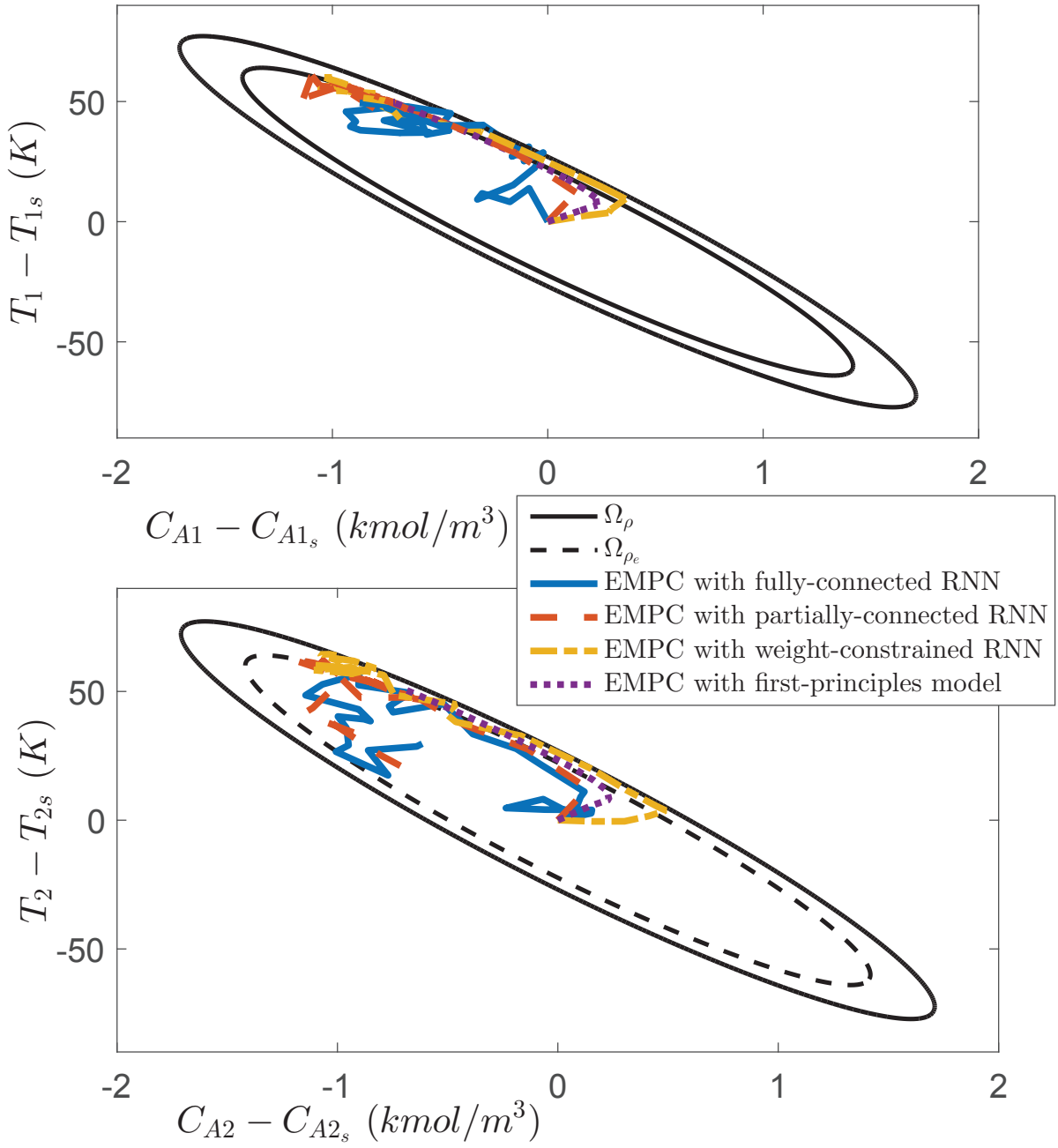


Figure 5.6: The state-space profiles for the closed-loop simulation for CSTR 1 (top plot) and CSTR 2 (bottom plot) under the EMPC using the fully-connected RNN model, the partially-connected RNN model, the weight-constrained RNN model, and the first-principles model of Eq. 5.13, respectively, for an initial condition (0, 0, 0, 0).

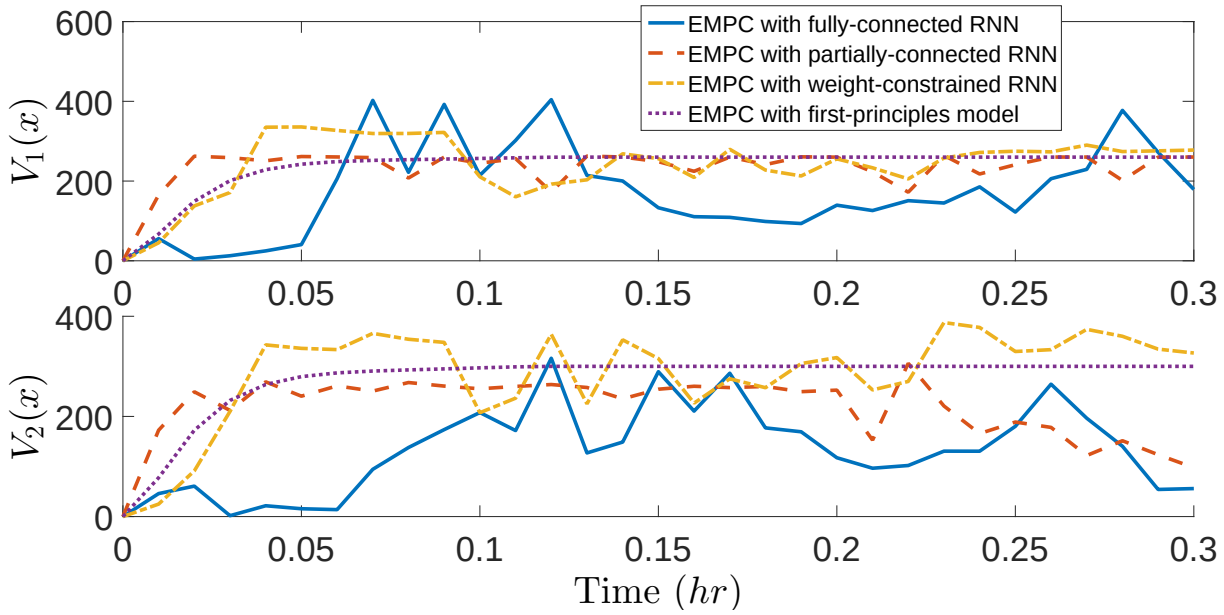


Figure 5.7: The Lyapunov function value evaluation with respect to time for the closed-loop CSTR 1 (top plot) and CSTR 2 (bottom plot) under the EMPC using the fully-connected RNN model, the partially-connected RNN model, the weight-constrained RNN model, and the first-principles model of Eq. 5.13, respectively, for an initial condition (0, 0, 0, 0).

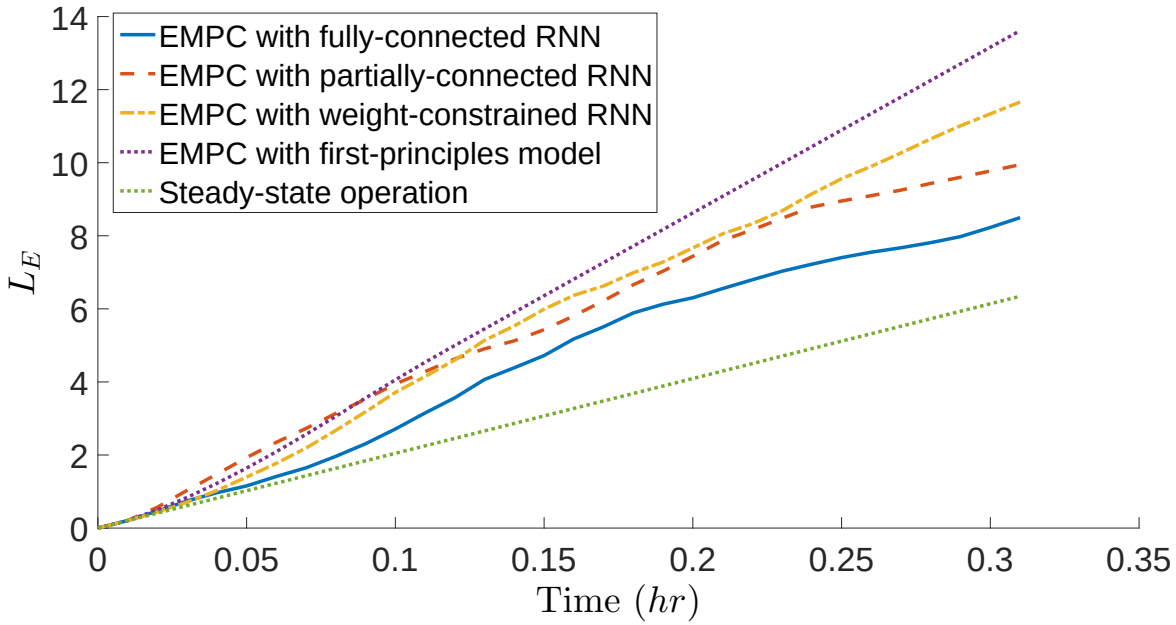


Figure 5.8: Accumulated economic profits for the closed-loop CSTRs under the steady-state operation and under the EMPC using the first-principles model of Eq. 5.13, the fully-connected RNN model, the partially-connected RNN model, and the weight-constrained RNN model, respectively, for an initial condition (0, 0, 0, 0).

Chapter 6

Control Lyapunov-Barrier Function-Based MPC

As discussed in the Introduction, maintaining safe and stable operation is the highest priority of the control systems in many safety-critical processes in chemical industries. In this chapter, novel MPC designs that take advantage of barrier functions and Lyapunov functions to ensure simultaneous closed-loop stability and process operational safety as well as recursive feasibility are developed. Specifically, barrier function that is commonly used to enforce safety properties in the context of optimization-based safety-critical controllers is first introduced. Subsequently, a new function termed *control Lyapunov-barrier function (CLBF)* is designed by combining a control barrier function with a control Lyapunov function through weighted sum, for which a rigorous stability and safety analysis is presented. Based on the CLBF-based controller that guarantees simultaneous stability and safety of nonlinear systems, CLBF-based MPC is developed and applied to chemical process examples to demonstrate, evaluate, and analyze the closed-loop stability and safety properties of nonlinear systems.

6.1 Preliminaries

6.2 Notation

The set of real numbers is denoted by $\mathbf{R}$, and the set of nonnegative real numbers is denoted by $\mathbf{R}_+$. $\mathbf{R}^n$ is an n-dimensional real (Euclidean) space. The notation $|\cdot|$ is used to denote the Euclidean norm of a vector, and the notation $|\cdot|_Q$ denotes a weighted Euclidean norm of a vector (i.e., $|x|_Q = \sqrt{x^T Q x}$ where Q is a positive definite matrix). x^T denotes the transpose of x. The notation $L_f V(x)$ denotes the standard Lie derivative of function $V(x)$ with respect to the vector field f, i.e., $L_f V(x) := \frac{\partial V(x)}{\partial x} f$. A scalar continuous function $V : \mathbf{R}^n \to \mathbf{R}$ is proper if the set $\{x \in \mathbf{R}^n \mid V(x) \leq k\}$ is compact for all $k \in \mathbf{R}$, or equivalently, V is radially unbounded in the sense that $\lim_{|x|\to+\infty} V(x) = +\infty$ holds.

For given positive real numbers β and ε, $\mathscr{B}_\beta(\varepsilon) := \{x \in \mathbf{R}^n \mid |x - \varepsilon| < \beta\}$ is an open ball around ε with radius of β. The relative complement of the set A in B is denoted by $A \backslash B := \{x \in A, x \notin B\}$. A function $f(\cdot)$ is of class $\mathscr{C}^1$ if it is continuously differentiable. A continuous function $\alpha : [0, a) \to \mathbf{R}_+$ is said to be of class $\mathscr{K}$ if it is strictly increasing and $\alpha(0) = 0$.

6.2.1 Class of Nonlinear Systems

The class of systems considered is described by the following system of nonlinear ordinary differential equations (ODEs):

$$\dot{x} = f(x) + g(x)u + h(x)w,\ x(t_0) = x_0 \tag{6.1}$$

where $x \in D \subset \mathbf{R}^n$ is the state vector, $u \in U \subset \mathbf{R}^m$ is the manipulated input vector, and $w \in W$ is the disturbance vector, where $W := \{w \in \mathbf{R}^l \mid |w| \leq \theta,\ \theta \geq 0\}$. The control action constraint is defined by $u \in U := \{u_{\min} \leq u \leq u_{\max}\} \subset \mathbf{R}^m$, where u_{min} and u_{max} are the lower and upper bounds for the input vector, respectively. It is assumed that $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are sufficiently smooth vector and matrix functions of dimensions $n \times 1$, $n \times m$, and $n \times l$, respectively, with $f(0) = 0$. Therefore,

the origin is a steady-state of the nominal system of Eq. 6.1 with $w(t) \equiv 0$. The measurement of $x(t)$ is assumed to be available for feedback at each sampling time $t_k = t_0 + k\Delta$, $k = 0, 1, \ldots$, where Δ is the sampling period. It is noted that we consider the control-affine nonlinear system in the form of Eq. 6.1 to simplify the discussion of explicit stabilizing controller design; however, the control Lyapunov-barrier function design and its incorporation in MPC/EMPC are not restricted to systems of Eq. 6.1, and can be generalized to the class of continuous-time nonlinear system in a general form:

$$\dot{x} = f(x, u, w) \tag{6.2}$$

where $f : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^l \to \mathbf{R}^n$ is a smooth vector function of its arguments with $f(0, 0, 0) = 0$.

6.2.2 Characterization of Unsafe Regions

We assume that there is a set $\mathscr{D}$ in state-space within which it is unsafe for the system to be operated, and a safe operating region $\mathscr{U}$ that has no intersection with $\mathscr{D}$, i.e., $\mathscr{U} \cap \mathscr{D} = \emptyset$. The definition of process operational safety for the closed-loop system of Eq. 6.1 is presented below.

Definition 6.1. *Consider the nominal system of Eq. 6.1 with $w(t) \equiv 0$ and input constraints $u \in U$. If there exists a control law $u = \Phi(x) \in U$ such that the state trajectories of the system for any initial state $x(t_0) = x_0 \in \mathscr{U}$ satisfy $x(t) \in \mathscr{U}$, $\forall\, t \geq t_0$, we say that the control law $\Phi(x)$ maintains the process state within the safe operating region $\mathscr{U}$ at all times.*

To ensure process operational safety, the unsafe region $\mathscr{D}$ should be first characterized by analyzing the safeness of processes based on first-principles models and past operating data. Generally, there are two types of unsafe regions: 1) unbounded sets, for example, an unsafe region consisting of all the states above a threshold that indicates an unsafe operation, and 2) bounded sets, which can be characterized based on multiple process states accounting for their interaction (e.g., a combination of temperature and concentration of reactants that reflect reaction rates in a chemical process example). Bounded unsafe sets are also commonly used in motion planning for robots and self-driving cars, which can be found, for example, in [95]. In this chapter, we will

address both bounded unsafe regions (denoted by $\mathscr{D}_b$) and unbounded unsafe regions (denoted by $\mathscr{D}_u$) and demonstrate that simultaneous closed-loop stability and process operational safety for the nonlinear system of Eq. 6.1 can be achieved under CLBF-based controllers.

6.3 Control Barrier Function

Consider the unforced nonlinear systems described by the following system of first-order nonlinear ordinary differential equations (ODEs):

$$\dot{x} = f(x) \tag{6.3}$$

where $x \in D \subset \mathbf{R}^n$, and $f : D \to \mathbf{R}^n$ is a smooth function of x. Barrier certificates were proposed in [121, 122] to ensure safety for the nonlinear system of Eq. 6.3 in the sense that the system is able to avoid undesirable regions. Since then, they have been successfully applied to solve safety critical control problems for cyber-physical systems, for example, obstacle avoidance problems for autonomous vehicles and collisions-free multi-robot systems [32, 159] Specifically, given a safe operation region $\mathscr{U}$ in state-space, there are two types of barrier certificates/functions $B(x)$ that are commonly used: one is the reciprocal barrier function that satisfies $B(x) \to \infty$ as $x \to \partial\mathscr{U}$, where $\partial\mathscr{U}$ represents the boundary of $\mathscr{U}$, and the other one is termed zeroing barrier function where $B(x) \to 0$ as $x \to \partial\mathscr{U}$. To ensure process safety for the system of Eq. 6.3, we show that the safe operating region $\mathscr{U}$ defined as a superlevel set [*] of a $\mathscr{C}^1$ function $r : D \to \mathbf{R}$ that satisfies the following conditions [12, 13]:

$$\mathscr{U} = \{x \in D \subset \mathbf{R}^n \mid r(x) \geq 0\}, \tag{6.4a}$$

$$\partial\mathscr{U} = \{x \in D \subset \mathbf{R}^n \mid r(x) = 0\}, \tag{6.4b}$$

$$Int(\mathscr{U}) = \{x \in D \subset \mathbf{R}^n \mid r(x) > 0\}, \tag{6.4c}$$

[*] $\{x \in \mathbf{R}^n \mid f(x) \geq c\}$ is called a superlevel set of $f : \mathbf{R}^n \to \mathbf{R}$, where c is a constant.

is an invariant set, where Int($\mathscr{U}$) represents the interior of the set $\mathscr{U}$. Specifically, we design a reciprocal barrier function as $B(x) = -log(\frac{r(x)}{1+r(x)})$ and impose a condition on the time-derivative of $B(x)$: $\dot{B} \leq \frac{\gamma}{B}$, where $\gamma > 0$. It is readily shown that the following conditions hold for $B(x)$:

$$\inf_{x \in Int(\mathscr{U})} B(x) > 0, \quad \lim_{x \to \partial \mathscr{U}} B(x) = \infty \tag{6.5}$$

Additionally, we derive the following inequality for $\dot{r}$ by differentiating $B(x)$ and using the constraint $\dot{B} \leq \frac{\gamma}{B}$:

$$\dot{r} \geq \frac{\gamma(h+h^2)}{log(\frac{h}{1+h})} \tag{6.6}$$

Using the Comparison Lemma [72], it is demonstrated that for any $r(x_0) > 0$, $r(x(t)) > 0$ holds for $t \geq 0$. Therefore, for any initial condition $x_0 \in \mathscr{U}$, the state remains inside $\mathscr{U}$ for all $t \geq 0$ (see [13] for the detailed proof).

Inspired by control Lyapunov functions (CLF) that were proposed for the nonlinear system with control inputs (e.g., Eq. 6.1) based on Lyapunov function for the unforced system of Eq. 6.3, barrier function was also extended to control barrier function (CBF) for the nonlinear affine control system of Eq. 6.1 with $w(t) \equiv 0$ in [165]. The definition of a CBF in [165] is presented below. For a comprehensive review on CBFs, the reader is referred to the review [11].

Definition 6.2. *Given a set of unsafe points $\mathscr{D}$ in state-space, a $\mathscr{C}^1$ function $B(x) : \mathbf{R}^n \to \mathbf{R}$ is a CBF if it satisfies the following properties:*

$$B(x) > 0, \quad \forall\, x \in \mathscr{D} \tag{6.7a}$$

$$L_f B(x) \leq 0, \; \forall\, x \in \{z \in \mathbf{R}^n \backslash \mathscr{D} \mid L_g B(z) = 0\} \tag{6.7b}$$

$$\mathscr{U} := \{x \in \mathbf{R}^n \mid B(x) \leq 0\} \neq \emptyset \tag{6.7c}$$

Additionally, a number of recent works, e.g., [133, 165, 185] have developed control laws for which they have been able to guarantee that the control law maintains safe operation of the process at all times when a CBF can be found for the system. The following theorem provides sufficient

conditions under which the existence of a CBF of Eq. 6.7 for the nominal system of Eq. 6.1 with $w(t) \equiv 0$ under the control law $u = \Phi_b(x)$ of Eq. 6.8 guarantees process operational safety of the closed-loop system for any initial condition $x_0 \in \mathscr{U}$.

Theorem 6.1. *Assume that the nominal system of Eq. 6.1 (i.e., $w(t) \equiv 0$) with no constraints on the control input u has a $\mathscr{C}^1$ CBF $B(x) : \mathbf{R}^n \to \mathbf{R}$ associated with an unsafe region $\mathscr{D}$ in state-space. The control law of Eq. 6.8 guarantees that the closed-loop state is bounded in the safe region $\mathscr{U}$ for all times if the initial condition x_0 is in $\mathscr{U}$.*

$$\Phi_b(x) = \begin{cases} -\dfrac{p + \sqrt{p^2 + \gamma |q|^4}}{|q|^2} q_i & \text{if} \quad q \neq 0 \\ 0 & \text{if} \quad q = 0 \end{cases} \tag{6.8}$$

where p denotes $L_f B(x)$, q_i denotes $L_{g_i} B(x)$, $q = [q_1 \cdots q_m]^T$, $f = [f_1 \cdots f_n]^T$, $g_i = [g_{i1} \cdots g_{in}]^T$, $(i = 1, 2, \cdots, m)$ and $\gamma > 0$.

Proof. By substituting the control law $u = \Phi_b(x)$ into the closed-loop system of Eq. 6.1, we can derive the following equation:

$$\dot{B}(x) = \frac{\partial B}{\partial x}(f(x) + g(x)\Phi_b(x)) = \begin{cases} -\sqrt{p^2 + \gamma |q|^4} & \text{if} \quad q \neq 0 \\ p & \text{if} \quad q = 0 \end{cases} \tag{6.9}$$

Since the CBF $B(x)$ satisfies Eq. 6.7b showing that $p \leq 0$ holds for all $x \in \mathbf{R}^n \backslash \mathscr{D}$ when $q = 0$, $\dot{B}(x)$ in Eq. 6.9 is guaranteed to be nonpositive for all $x \in \mathbf{R}^n \backslash \mathscr{D}$. Therefore, if the state starts from $\mathscr{U} \subset (\mathbf{R}^n \backslash \mathscr{D})$, the value of $B(x)$ is guaranteed to be non-increasing along the trajectory of x. This completes the proof that the safe operating region $\mathscr{U}$ is an invariant set under $u = \Phi_b(x)$. □

6.4 Control Lyapunov-Barrier Function

To address simultaneously the tasks of stability, safety, and other considerations such as economic optimality, control Lyapunov functions (CLF) and control barrier functions (CBF) are utilized

to design process control systems. Specifically, process operational safety in the sense that the state is bounded in a safe operating region is guaranteed under the CBFs satisfying Lyapunov-like conditions [109, 145, 165]. CBFs can be naturally unified with CLFs to formulate a quadratic program, which allows for the satisfaction of the objectives of stability and safety (see, for example, [12, 13, 65]). Additionally, another approach to solving the problem of stabilization of a nonlinear process with guaranteed safety is to use control Lyapunov-barrier functions (CLBF), which is a function that combines CBFs and CLFs via weighted sum.

In this section, we introduce the definition of CLBFs for input-constrained system of Eq. 6.1, followed by the construction method that separates the control design for achieving the asymptotic stability and safety by designing the CLF and CBF, independently, and then combines them together [133]. The design of CLBF-based controller for the nonlinear system of Eq. 6.1 will be discussed with a rigorous theoretical analysis on closed-loop stability and operational safety.

6.4.1 Stabilization and Safety via Control Lyapunov-Barrier Function

6.4.1.1 Stabilizability assumptions

Assumption 6.1. *We assume that the nominal system of Eq. 6.1 with $w(t) \equiv 0$ admits a positive definite and proper control Lyapunov function V that satisfies the following condition:*

$$L_f V(x) < 0, \forall x \in \{z \in \mathbf{R}^n \backslash \{0\} \mid L_g V(z) = 0\} \tag{6.10}$$

We also assume that V satisfies the small control property, i.e., for every $\varepsilon > 0$, $\exists\ \delta > 0$, s.t. $\forall\ x \in \mathscr{B}_\delta(0)$, there exists u that satisfies $|u| < \varepsilon$ and $L_f V(x) + L_g V(x) u < 0$.

The CLF assumption implies the existence of a stabilizing feedback control law $\Phi(x)$ that renders the origin asymptotically stable in the sense that Eq. 6.10 holds for $u = \Phi(x)$, where $\Phi(x) \in U$. An example of a feedback control law that is continuous for all x in a neighborhood of the origin and renders the origin asymptotically stable is the following control law [83]:

$$k_i(x) = \begin{cases} -\dfrac{p+\sqrt{p^2+\gamma|q|^4}}{|q|^2} q_i & \text{if} \quad q \neq 0 \\ 0 & \text{if} \quad q = 0 \end{cases} \tag{6.11a}$$

$$\Phi_i(x) = \begin{cases} u_{min} & \text{if} \quad k_i(x) < u_{min} \\ k_i(x) & \text{if} \quad u_{min} \leq k_i(x) \leq u_{max} \\ u_{max} & \text{if} \quad k_i(x) > u_{max} \end{cases} \tag{6.11b}$$

where p denotes $L_f V(x)$, q_i denotes $L_{g_i} V(x)$, $q = [q_1 \cdots q_m]^T$, $f = [f_1 \cdots f_n]^T$, $g_i = [g_{i1} \cdots g_{in}]^T$, $(i = 1, 2, \cdots, m)$ and $\gamma > 0$. $k_i(x)$ of Eq. 6.11a represents the i_{th} component of the control law $\Phi(x)$ before considering saturation of the control action at the input bounds. $\Phi_i(x)$ of Eq. 6.11b represents the i_{th} component of the saturated control law $\Phi(x)$ that accounts for the input constraint $u \in U$.

6.4.1.2 Stabilization and safety via CLBF

Control Lyapunov-barrier function (CLBF) was originally proposed in [133], where the stabilization and safety-related results were guaranteed only for $u \in \mathbf{R}^m$ (i.e., no input constraints). Considering the fact that practical nonlinear systems are often subject to input constraints $u \in U$ as assumed in Eq. 6.1, a new CLBF must be developed to derive similar results. Therefore, based on the definition of a CLBF in [133], we propose a modified CLBF (termed constrained CLBF or simply CLBF in this chapter) that accounts for the presence of input constraints in the system of Eq. 6.1. Specifically, the definition of a constrained CLBF is as follows:

Definition 6.3. *Given a set of unsafe points $\mathscr{D}$ in state-space, a proper, lower-bounded and $\mathscr{C}^1$ function $W_c(x) : \mathbf{R}^n \to \mathbf{R}$ is a constrained CLBF if $W_c(x)$ has a minimum at the origin and also*

satisfies the following properties:

$$W_c(x) > \rho_c, \ \ \forall\, x \in \mathscr{D} \subset \phi_{uc} \tag{6.12a}$$

$$L_f W_c(x) < 0, \ \ \forall\, x \in \{z \in \phi_{uc} \backslash (\mathscr{D} \cup \{0\} \cup \mathbb{X}_e) \mid L_g W_c(z) = 0\} \tag{6.12b}$$

$$\mathscr{U}_{\rho_c} := \{x \in \phi_{uc} \mid W_c(x) \leq \rho_c\} \neq \emptyset \tag{6.12c}$$

$$\overline{\phi_{uc} \backslash (\mathscr{D} \cup \mathscr{U}_{\rho_c})} \cap \overline{\mathscr{D}} = \emptyset \tag{6.12d}$$

where $\rho_c \in \mathbf{R}$, ϕ_{uc} is a neighborhood around the origin, and $\mathbb{X}_e := \{x \in \phi_{uc} \backslash (\mathscr{D} \cup \{0\}) \mid \frac{\partial W_c(x)}{\partial x} = 0\}$ is a set of states where $L_f W_c(x) = 0$ (for $x \neq 0$) due to $\partial W_c(x)/\partial x = 0$.

We assume that there exists a feedback control law $u = \Phi(x) \in U$ (e.g., the universal Sontag control law of Eq. 6.11 with $W_c(x)$ replacing $V(x)$) such that the state of the closed-loop nominal system of Eq. 6.1 is bounded in a level set of $W_c(x)$ in an open neighborhood D_0 that includes the origin in its interior in the sense that there exists a $\mathscr{C}^1$ constrained CLBF $W_c(x)$ that has a minimum at the origin and satisfies the following inequalities for all $x \in D_0$:

$$\alpha_1(|x|) \leq W_c(x) - \rho_0 \leq \alpha_2(|x|), \tag{6.13a}$$

$$\begin{aligned} &\frac{\partial W_c(x)}{\partial x} F(x, \Phi(x), 0) \leq -\alpha_3(|x|), \forall x \in D_0 \backslash \mathscr{B}_\delta(x_e) \\ &\frac{\partial W_c(x)}{\partial x} F(x, \Phi(x), 0) \leq 0, \forall x \in \mathscr{B}_\delta(x_e) \end{aligned} \tag{6.13b}$$

$$\left| \frac{\partial W_c(x)}{\partial x} \right| \leq \alpha_4(|x|) \tag{6.13c}$$

where $\alpha_j(\cdot)$, $j = 1,\ 2,\ 3,\ 4$ are class $\mathscr{K}$ functions, $W_c(0) = \rho_0$ is the global minimum value of $W_c(x)$ in D_0, and $\mathscr{B}_\delta(x_e)$ is a small neighborhood around the stationary points $x_e \in \mathbb{X}_e$. $F(x,u,w)$ is used to represent the system of Eq. 6.1 (i.e., $F(x,u,w) := f(x) + g(x)u + h(x)w$). It is noted that in Eq. 6.13b, $\frac{\partial W_c(x)}{\partial x} F(x, \Phi(x), 0) \leq -\alpha_3(|x|)$ does not hold for all $x \in \mathscr{B}_\delta(x_e)$ since $\frac{\partial W_c(x)}{\partial x}$ is close to zero in the neighborhood around x_e, where $\frac{\partial W_c(x)}{\partial x} = 0$. Additionally, by continuity and the smoothness properties assumed for f, g and h in the nonlinear system of Eq. 6.1, there exist positive

constants $M, L_x, L_w, L_x^{'}, L_w^{'}$ such that the following inequalities hold for all $x, x' \in \mathscr{U}_{\rho_c} \subset D_0, u \in U$, and $w \in W$:

$$|F(x,u,w)| \leq M \tag{6.14a}$$

$$|F(x,u,w) - F(x',u,0)| \leq L_x|x-x'| + L_w|w| \tag{6.14b}$$

$$\left| \frac{\partial W_c(x)}{\partial x} F(x,u,w) - \frac{\partial W_c(x')}{\partial x} F(x',u,0) \right| \leq L_x^{'}|x-x'| + L_w^{'}|w| \tag{6.14c}$$

Based on the stabilizability and safety requirements of Eq. 6.13, we can find a positive real number a such that $\{x \in D_0 \mid \dot{W}_c(x(t)) = L_f W_c + L_g W_c u < -a|W_c(x) - W_c(0)|\}$ is not an empty set under the stabilizing control law $u = \Phi(x) \in U$ (e.g., the Sontag control law of Eq. 6.11 with $W_c(x)$ replacing $V(x)$). Therefore, ϕ_{uc} is defined to be the union of the above set, the origin, and $\mathscr{B}_\delta(x_e)$, i.e., $\phi_{uc} := \{x \in D_0 \mid \dot{W}_c(x(t)) < -a|W_c(x) - W_c(0)|, u = \Phi(x) \in U\} \cup \{0\} \cup \mathscr{B}_\delta(x_e)$. Additionally, we define the set of initial conditions by $\mathbb{X}_0 := \{x \in \phi_{uc} \backslash \mathscr{D}\}$ where $(\{0\} \cup \mathbb{X}_e) \in \mathbb{X}_0$, and thus, it is readily shown that the set $\mathscr{U}_{\rho_c}$ defined by Eq. 6.12c is a subset of $\mathbb{X}_0$. From now on, we will denote $\dot{W}_c(x(t))$, if not otherwise stated, simply by $\dot{W}_c$.

6.4.1.3 Closed-loop stability and safety under CLBF-based controller

We analyze closed-loop stability and safety for the following two cases: a bounded unsafe region $\mathscr{D}_b$ and an unbounded unsafe region $\mathscr{D}_u$ in state-space. The definition of simultaneous closed-loop stability and operational safety for the nonlinear system of Eq. 6.1 is presented below.

Definition 6.4. *Consider the nominal system of Eq. 6.1 with $w(t) \equiv 0$ and input constraints $u \in U$. If for any initial state $x(t_0) = x_0 \in \mathscr{U}$, there exists a control action $u \in U$ such that the state trajectories of the closed-loop system satisfy $x(t) \in \mathscr{U}$, $\forall\, t \geq t_0$, and $\lim_{t\to\infty} |x(t)| \leq d$, where $\mathscr{B}_d(0)$ is a small neighborhood around the origin, then we say that closed-loop stability and operational safety are achieved simultaneously in the sense that the process state is maintained within a safe operating region at all times, and can be ultimately driven to the origin.*

Case 1: If the unsafe region is characterized as a bounded set $\mathscr{D}_b$, it has been demonstrated

in [23] that asymptotic stability of the origin cannot be achieved under the continuous control law $u = \Phi(x) \in U$ due to the existence of other stationary points (i.e., $x_e \in \mathbb{X}_e$ and $x_e \neq 0$). In other words, for some $x_0 \in \mathbb{X}_0$, the closed-loop state may be trapped in x_e (such stationary points x_e can be either local minima or saddle points of $W_c(x)$) instead of the origin which has the global minimum of $W_c(x)$ under $u = \Phi(x)$. Specifically, as shown in Fig. 6.1, since there exist initial states $x_0 \in \mathscr{U}_{\rho_c} \subset \phi_{uc}$ such that the trajectories from x_0 pass around $\mathscr{D}_b$ in all possible directions, a discontinuous control action has to be applied at x_e to choose a direction to drive the state around $\mathscr{D}_b$ and towards the origin. Moreover, it is noted that in order to escape from x_e and converge to the origin, $W_c(x)$ needs to be carefully designed (e.g., the shapes and functional forms of $W_c(x)$) such that x_e is a saddle point rather than a local minimum. Since x_e can be characterized once the form of $W_c(x)$ is determined, a set of control actions $\bar{u}$ that can drive the state away from the saddle point in the direction of decreasing $W_c(x)$ should also be calculated in advance and be applied when the closed-loop state converges to x_e.

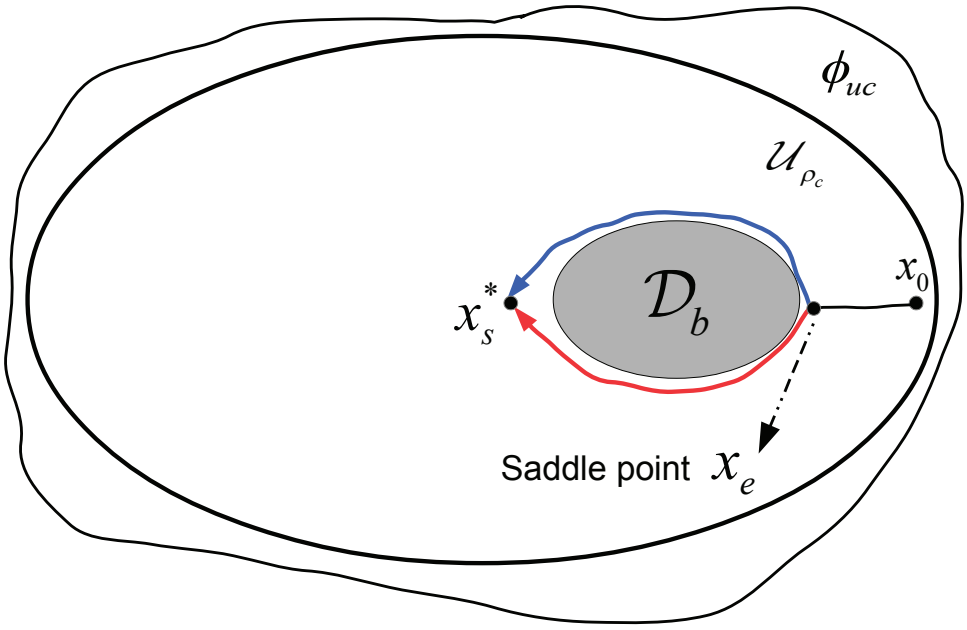


Figure 6.1: A schematic representing a bounded unsafe set $\mathscr{D}_b$ embedded within the operating region, where there exists an initial condition x_0 and a saddle point x_e such that the trajectories from x_0 converge to x_e and pass around $\mathscr{D}_b$ either in the up or down direction with a discontinuous control action at x_e.

Theorem 6.2 below provides sufficient conditions under which the existence of a constrained

CLBF of Eq. 6.12 for the nominal system of Eq. 6.1 with $w(t) \equiv 0$ under the control law $\Phi(x)$ guarantees that the solution of the system of Eq. 6.1 always stays in a safe operating region. The proof of the theorem follows from the results in [169, 172].

Theorem 6.2. *Consider that a constrained CLBF $W_c(x) : \mathbf{R}^n \to \mathbf{R}$, that has a minimum at the origin, exists for the nominal system of Eq. 6.1 (i.e., $w(t) \equiv 0$) with the input constraints $u \in U$, defined with respect to a bounded unsafe region $\mathscr{D}_b$ in state-space. The feedback control law $\Phi(x)$ guarantees that the closed-loop state stays in $\mathbb{X}_0$ and does not enter $\mathscr{D}_b$ for all times for $x(0) = x_0 \in \mathbb{X}_0$.*

Proof. First, we prove that if $x_0 \in \mathbb{X}_0$ where $\mathbb{X}_0 := \{x \in \phi_{uc} \backslash \mathscr{D}_b\}$, then the closed-loop state will never enter $\mathscr{D}_b$, for all $t \geq 0$. Consider the first case that $x_0 \in \mathscr{U}_{\rho_c} \subset \mathbb{X}_0$. By the definition of ϕ_{uc}, it is guaranteed that $\dot{W}_c$ is negative everywhere in the set $\mathbb{X}_0 \backslash (\{0\} \cup \mathbb{X}_e)$. (e.g., if $L_g W_c(x) = 0$, it follows that $\dot{W}_c(x) = L_f W_c(x) < 0$; if $L_g W_c(x) \neq 0$, it follows that $\dot{W}_c = -\sqrt{L_f W_c^2 + \gamma |L_g W_c|^4} < 0$ using the Sontag control law of Eq. 6.11 with $W_c(x)$ replacing $V(x)$). Additionally, if $x \in \mathbb{X}_e$, $\dot{W}_c(x) = 0$ holds. Therefore, it follows that $W_c(x(t)) \leq W_c(x(0))$ for all $x(t) \in \mathscr{U}_{\rho_c}$ by $\dot{W}_c \leq 0$, i.e., $x(t)$ stays in the set $\mathscr{U}_{\rho_c}$ for all $t \geq 0$ if $x_0 \in \mathscr{U}_{\rho_c}$.

Also, $\mathscr{U}_{\rho_c}$ is a compact invariant set due to the properness of W_c and the property $\dot{W}_c \leq 0$. Due to the fact that $\mathscr{U}_{\rho_c} \cap \mathscr{D}_b = \emptyset$, it follows that for any $x_0 \in \mathscr{U}_{\rho_c}$, the closed-loop state does not enter the set of unsafe states at any time (i.e., it is maintained within the set of safe states at all times). Additionally, since any subset of $\mathscr{U}_{\rho_c}$, $\mathscr{U}_\rho := \{x \in \phi_{uc} \mid W_c(x) \leq \rho\}$ where $\rho \leq \rho_c$, is also a compact invariant set, we can show that if $x_0 \in \mathscr{U}_\rho$, it holds that $x(t) \in \mathscr{U}_\rho$, $\forall t \geq 0$. It remains to be shown that for all other initial states $x_0 \in \phi_{uc} \backslash (\mathscr{D}_b \cup \mathscr{U}_{\rho_c})$, $x(t) \notin \mathscr{D}_b, \forall\, t \geq 0$. Given an initial state x_0 that belongs to the set $\phi_{uc} \backslash (\mathscr{D}_b \cup \mathscr{U}_{\rho_c})$, $W_c(x_0) > \rho_c$ holds because it is not within the set $\mathscr{U}_{\rho_c}$ defined in Eq. 6.12c. However, since Eq. 6.12b holds within $\phi_{uc} \backslash (\mathscr{D}_b \cup \{0\})$, the conclusion that $\dot{W}_c(x)$ is negative along the trajectory of $x(t)$ holds using the same steps as performed above when x_0 was within $\mathscr{U}_{\rho_c}$. Furthermore, since the set $\overline{\phi_{uc} \backslash (\mathscr{D}_b \cup \mathscr{U}_{\rho_c})}$ does not intersect with $\overline{\mathscr{D}_b}$, any trajectory starting in $\phi_{uc} \backslash (\mathscr{D}_b \cup \mathscr{U}_{\rho_c})$ will reach the boundary of $\phi_{uc} \backslash (\mathscr{D}_b \cup \mathscr{U}_{\rho_c})$ before reaching the boundary of $\mathscr{D}_b$. Because Eq. 6.12d holds (i.e., $\overline{\phi_{uc} \backslash (\mathscr{D}_b \cup \mathscr{U}_{\rho_c})} \cap \overline{\mathscr{D}_b} = \emptyset$), it must hold that

$\overline{\phi_{uc}\backslash(\mathscr{D}_b \cup \mathscr{U}_{\rho_c})} \cap \mathscr{U}_{\rho_c}$, is a nonempty set. Because $W_c(x) > \rho_c$ within $\phi_{uc}\backslash(\mathscr{D}_b \cup \mathscr{U}_{\rho_c})$ but $W_c(x) \leq \rho_c$ within $\mathscr{U}_{\rho_c}$ from Eq. 6.12c, $W_c(x) = \rho_c$, $\forall x \in \partial\phi_{uc}\backslash(\mathscr{D}_b \cup \mathscr{U}_{\rho_c})$ due to the continuity of W_c, which means that the trajectory will enter and remain in $\mathscr{U}_{\rho_c}$ after it reaches the boundary of $\phi_{uc}\backslash(\mathscr{D}_b \cup \mathscr{U}_{\rho_c})$. This completes the proof that for all $x_0 \in \mathbb{X}_0$, $x(t) \notin \mathscr{D}_b, \forall\, t \geq 0$. □

Remark 6.1. *In Theorem 6.2, simultaneous stability (boundedness of the closed-loop state) and safety are proved for the nominal system of Eq. 6.1 with any $x_0 \in \mathbb{X}_0$ under $u = \Phi(x)$. Note that the set of initial condition $\mathbb{X}_0$ contains two parts. One is $\mathscr{U}_{\rho_c}$ of Eq. 6.12c, where it satisfies $W_c(x) \leq \rho_c$; the other one is $\overline{\phi_{uc}\backslash(\mathscr{D}_b \cup \mathscr{U}_{\rho_c})}$, which is required to satisfy Eq. 6.12d. Therefore, if we restrict the initial conditions to $\mathscr{U}_{\rho_c}$ or any subset of it, the conditions of a constraint CLBF in Eq. 6.12 can be reduced to Eqs. 6.12a-6.12c. Otherwise, if the set $\overline{\phi_{uc}\backslash(\mathscr{D}_b \cup \mathscr{U}_{\rho_c})}$ is considered as a part of initial conditions, all the conditions in Eq. 6.12 are required to hold for W_c. The additional condition of Eq. 6.12d for the case of $x_0 \in \overline{\phi_{uc}\backslash(\mathscr{D}_b \cup \mathscr{U}_{\rho_c})}$ also implies that $W_c(x) = \rho_c$ for all $x \in \partial\mathscr{D}_b$, which can be readily shown by contradiction.*

Case 2: If an unbounded unsafe region $\mathscr{D}_u$ is considered, there does not exist such a stationary point $x_e \neq 0$ according to [23]. As a result, Eq. 6.12 can be simplified with $\mathbb{X}_e = \emptyset$ and $\dot{W}_c < 0$ holds for all $x \in \mathscr{U}_{\rho_c}\backslash\{0\}$ under the controller $u = \Phi(x) \in U$. It is shown in Fig. 6.2 that in this case, the trajectories from $x_0 \in \mathscr{U}_{\rho_c}$ converge to the origin while avoiding $\mathscr{D}_u$ in one direction. Additionally, from now on, we will restrict the set of initial conditions to be in $\mathscr{U}_{\rho_c}$ (i.e., $x_0 \in \mathscr{U}_{\rho_c}$) to simplify the discussion. The following theorem demonstrates that closed-loop stability and process operational safety are achieved simultaneously for the system of Eq. 6.1 under $u = \Phi(x) \in U$.

Theorem 6.3. *Consider that a constrained CLBF $W_c(x) : \mathbf{R}^n \to \mathbf{R}$ that has a minimum at the origin and meets the conditions of Eq. 6.12, exists for the nominal system of Eq. 6.1 with $w(t) \equiv 0$ subject to input constraints, defined with respect to an unbounded unsafe region $\mathscr{D}_u$ in state-space. The continuous feedback control law $u = \Phi(x) \in U$ guarantees that the closed-loop state is bounded in $\mathscr{U}_{\rho_c}$ for all times and the origin can be rendered asymptotically stable $\forall x_0 \in \mathscr{U}_{\rho_c}$.*

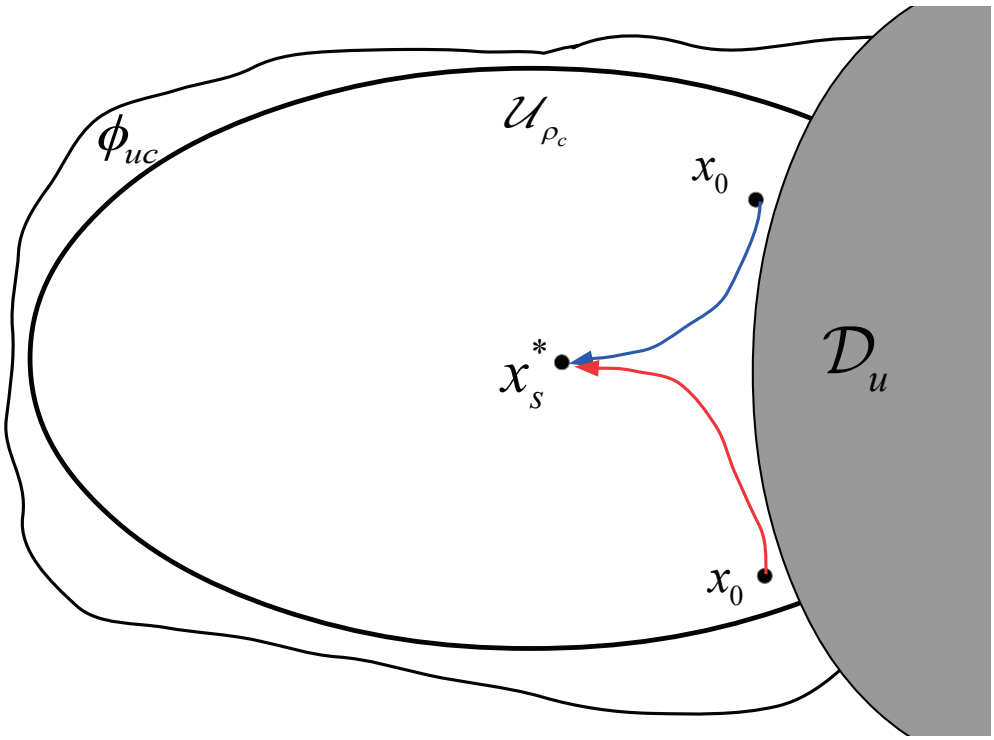


Figure 6.2: A schematic representing an unbounded unsafe set $\mathscr{D}_u$ in state-space, where the trajectories starting from any initial condition x_0 avoid $\mathscr{D}_u$ and converge to the origin x_s^*.

Proof. Following the first part of proof for Theorem 6.2, it is readily shown that $\dot{W}_c$ remains negative for all x in the set $\phi_{uc}\backslash\{0\}$ under the controller $u = \Phi(x) \in U$, which implies that $\forall x_0 \in \mathscr{U}_{\rho_c} \subset \phi_{uc}$, the state stays in $\mathscr{U}_{\rho_c}$ for all times and will ultimately converge to the origin due to the fact that $\dot{W}_c < 0, \forall x \in \mathscr{U}_{\rho_c}\backslash\{0\}$. □

Remark 6.2. *Control Lyapunov-barrier function $W_c(x)$ and Lyapunov function $V(x)$ are similar in that they both have a global minimum at the origin of state-space and the level sets of $W_c(x)$ and $V(x)$ are both invariant sets. However, the level sets of a CLBF can have negative upper bounds (i.e., $\rho_c < 0$) and there exist multiple stationary points (other than the origin) for $W_c(x)$. Thus, the Lyapunov-based control law (e.g., Sontag control law of Eq. 6.11 in terms of $V(x)$) guarantees convergence of the state to the origin (i.e., the equilibrium point at the origin is asymptotically stable), while the CLBF-based control law (e.g., Sontag control law of Eq. 6.11 in terms of $W_c(x)$) guarantees boundedness of the state and avoidance of the unsafe region in a level set of $W_c(x)$. Additionally, convergence of the state to the origin can be guaranteed for CLBF-based control law*

if other stationary points (i.e., saddle points) are addressed using a discontinuous control law.

6.4.2 Design of Constrained CLBF

The method for constructing a constrained CLBF is discussed in this section. Specifically, a constrained CLBF can be constructed by combining a CLF and a CBF that have been separately designed, and we present a practical method for designing a CLBF that satisfies the properties in Eq. 6.12. Proposition 6.1 below provides the guidelines for choosing the CLF and CBF, and the corresponding weights, through which the global minimum of $W_c(x)$ is achieved at the origin.

Proposition 6.1. *Given an open set $\mathscr{D}$ of unsafe states for the nominal system $\dot{x} = f(x) + g(x)u + h(x)w$ with $w(t) \equiv 0$, assume that there exists a $\mathscr{C}^1$ CLF $V : \mathbf{R}^n \to \mathbf{R}_+$, and a $\mathscr{C}^1$ CBF $B : \mathbf{R}^n \to \mathbf{R}$, such that the following conditions hold:*

$$c_1 |x|^2 \leq V(x) \leq c_2 |x|^2, \ \forall x \in \mathbf{R}^n, c_2 > c_1 > 0 \tag{6.15}$$

$$\mathscr{D} \subset H \subset \phi_{uc}, \ 0 \notin H \tag{6.16}$$

$$B(x) = -\eta < 0, \ \forall x \in \mathbf{R}^n \backslash H; \ B(x) > 0, \ \forall x \in \mathscr{D} \tag{6.17}$$

where H is a compact and connected set within ϕ_{uc}. Define $W_c(x)$ to have the form $W_c(x) := V(x) + \mu B(x) + \nu$, where:

$$L_f W_c(x) < 0, \ \ \forall \, x \in \{z \in \phi_{uc} \backslash (\mathscr{D} \cup \{0\} \cup \mathbb{X}_e) \mid L_g W_c(z) = 0\} \tag{6.18}$$

$$\mu > \frac{c_2 c_3 - c_1 c_4}{\eta}, \tag{6.19a}$$

$$\nu = \rho_c - c_1 c_4, \tag{6.19b}$$

$$c_3 := \max_{x \in \partial H} |x|^2, \tag{6.19c}$$

$$c_4 := \min_{x \in \partial \mathscr{D}} |x|^2. \tag{6.19d}$$

Then for initial states $x_0 \in \phi_{uc} \backslash \mathscr{D}_H$, where $\mathscr{D}_H := \{x \in H \mid W_c(x) > \rho_c\}$, the control law $\Phi(x)$ of Eq. 6.11 (with $W_c(x)$ replacing $V(x)$) guarantees that the closed-loop state is bounded in $\phi_{uc} \backslash \mathscr{D}_H$ and does not enter the unsafe region $\mathscr{D}_H$ for all times.

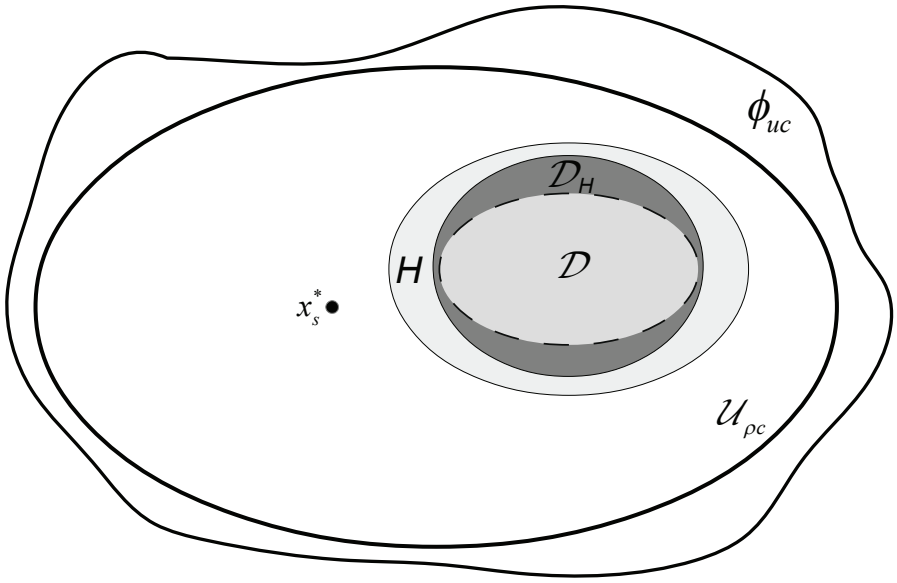


Figure 6.3: A schematic representing the relationship between the sets ϕ_{uc}, $\mathscr{D}$, $\mathscr{D}_H$ and H, where the invariant set $\mathscr{U}_{\rho_c}$ is shown as an ellipse subtracting $\mathscr{D}_H$.

Proof. We define a new compact and connected set H, which satisfies Eq. 6.16, and an expanded unsafe region $\mathscr{D}_H$, such that all the states with $W_c(x) > \rho_c$ inside the region H are included in $\mathscr{D}_H$. A schematic describing the above relationship among different sets is shown in Fig. 6.3. We prove that the proposed constrained CLBF, $W_c(x)$, meets all the requirements of Eq. 6.12 with $\mathscr{D}_H$

replacing $\mathscr{D}$ and has a global minimum at the origin. Firstly, it is trivial to show that Eq. 6.12a holds by the definition of $\mathscr{D}_H$. Additionally, we can use Eqs. 6.15, 6.17, and 6.19 to show that for all $x \in \mathscr{D}$, $W_c(x) > \rho_c$ also holds as follows:

$$\begin{aligned} W_c(x) &= V(x) + \mu B(x) + \nu \\ &> c_1 |x|^2 + \rho_c - c_1 c_4 \\ &> \rho_c \end{aligned} \tag{6.20}$$

Eq 6.12b is also trivially satisfied by the proposed CLBF via the required property of Eq. 6.18. To prove that Eq. 6.12c holds, we obtain the following inequalities for all $x \in \partial H$,

$$\begin{aligned} W_c(x) &= V(x) + \mu B(x) + \nu \\ &\leq c_2 |x|^2 - \mu\eta + \rho_c - c_1 c_4 \\ &< \rho_c \end{aligned} \tag{6.21}$$

Hence, Eq. 6.12c holds due to the fact that $\mathscr{U}_{\rho_c} \neq \emptyset$ obtained from Eq. 6.21, which also implies that $\partial H \cap \partial \mathscr{D}_H = \emptyset$. Following this, we have $\mathscr{D}_H \subset H \subset (\mathscr{D}_H \cup \mathscr{U}_{\rho_c})$, which implies the boundary of $\phi_{uc} \backslash (\mathscr{D}_H \cup \mathscr{U}_{\rho_c})$ does not intersect with the boundary of $\mathscr{D}_H$, (i.e., Eq. 6.12d holds, $\overline{\phi_{uc} \backslash (\mathscr{D}_H \cup \mathscr{U}_{\rho_c})} \cap \overline{\mathscr{D}_H} = \emptyset$). Additionally, $W_c(x)$ has a global minimum at the origin since the minimums of $V(x)$ and $B(x)$ are both at the origin. Therefore, we can conclude that for any initial states $x_0 \in \phi_{uc} \backslash \mathscr{D}_H$, the control law $\Phi(x)$ guarantees that the closed-loop state is bounded in $\phi_{uc} \backslash \mathscr{D}_H$ and does not enter $\mathscr{D}_H$ for all times. □

6.5 CLBF-based Model Predictive Control

In this section, a CLBF-based model predictive control (CLBF-MPC) scheme that incorporates CLBF-based stability and safety constraints is proposed to regulate the nonlinear system of Eq. 6.1 to the steady-state while avoiding the unsafe operation at the same time. We first provide a brief

overview of model predictive control methods. Then, we discuss the impact of sample-and-hold implementation of control actions on the stability and safety properties derived by the continuous controller $u = \Phi(x) \in U$. Subsequently, a rigorous theoretical treatment of the closed-loop stability and safety properties of the system of Eq. 6.1 with the control architecture is provided.

6.5.1 Sample-and-hold Implementation of CLBF-based Controller

As Lyapunov-based MPC only accounts for closed-loop stability, the avoidance of unsafe regions in state-space is not guaranteed. Therefore, a new design of MPC that incorporates CLBF constraints is developed in the following sections. In the proof of Theorem 6.2, it was noted that when a constrained CLBF exists for the nominal system of Eq. 6.1, i.e., $\dot{x} = f(x) + g(x)u + h(x)w$ with $w(t) \equiv 0$, the controller $\Phi(x)$ when continuously implemented, can maintain the state in a safe region of operation. Since the CLBF will be used to design constraints for MPC, for which control actions are implemented in sample-and-hold, the sample-and-hold properties of the controller $\Phi(x)$ (with a sampling period Δ) must be investigated in the presence of disturbances. The next proposition and its proofs develop these results.

Proposition 6.2. *Consider the nominal system of Eq. 6.1 with a constrained CLBF W_c that meets the requirements of Definition 6.3 and has a minimum at the origin, and the set of initial conditions $\mathscr{U}_{\rho_c} \subset \mathbb{X}_0$. Let $u(t) = \Phi(x(t_k))$, $t_k \leq t < t_{k+1}$, for any $x(t_k) \in \mathscr{U}_{\rho_c} \backslash \mathscr{B}_\delta(x_e)$ where $\delta > 0$, $x_e \in \mathbb{X}_e$ and t_k represents the time instance, i.e., $t_k = k\Delta$, $k = 0, 1, 2, \ldots$. Let $u(t) = \bar{u}(x) \in U$ such that if $x(t_k) \in \mathscr{B}_\delta(x_e)$, $W_c(x(t_{k+1})) < W_c(x(t_k))$ holds for any $\Delta > 0$ under $\bar{u}(x)$. Then, given any positive real number d, there exists a positive real number Δ^*, such that, if $\Delta \in (0, \Delta^*]$ and $x_0 \in \mathscr{U}_{\rho_c}$, then $x(t) \in \mathscr{U}_{\rho_c}$, $\forall t \geq 0$, and $\lim_{t\to\infty} |x(t)| \leq d$.*

Proof. We need to show that under sample-and-hold implementation, any states originating in $\mathscr{U}_{\rho_c}$ converge to a level set around the origin $\mathscr{U}_{\rho_{min}} := \{x \in \phi_{uc} \mid W_c(x) \leq \rho_{min}\}$ as $t \to \infty$ where $\rho_{min} < \rho_c$. Following this, it is trivial to show that $x(t) \in \mathscr{U}_{\rho_{min}}$ as $t \to \infty$ implies $\lim_{t\to\infty} |x(t)| \leq d$ by the continuity of $W_c(x)$. To prove that the state will converge to $\mathscr{U}_{\rho_{min}}$, we first show that

$\forall\, x(t_k) \in \mathscr{U}_{\rho_c} \backslash (\mathscr{U}_{\rho_s} \cup \mathscr{B}_\delta(x_e))$, where $\rho_s < \rho_{min} < \rho_c$, $\dot{W}_c(x(t),u(t)) < -\varepsilon$ holds in the set $\mathbb{Z} := \{x \in \phi_{uc} \backslash \mathscr{B}_\delta(x_e) \mid \rho_s \leq W_c(x) \leq \rho_c\}$ with $u(t) = u(t_k) = \Phi(x(t_k))$, $\forall\, t \in [t_k, t_k + \Delta^*)$ as below:

$$\begin{aligned}\dot{W}_c(x(t),u(t)) =& \dot{W}_c(x(t_k),u(t_k)) + (\dot{W}_c(x(t),u(t)) - \dot{W}_c(x(t_k),u(t_k))) \\ =& L_f W_c(x(t_k)) + L_g W_c(x(t_k))u(t_k) + (L_f W_c(x(t)) - L_f W_c(x(t_k))) \\ &+ (L_g W_c(x(t)) - L_g W_c(x(t_k)))u(t)\end{aligned} \tag{6.22}$$

where $\dot{W}_c(x,u)$ is used to represent $\frac{\partial W_c(x)}{\partial x}(f(x)+g(x)u)$. Due to the smoothness of $f(\cdot)$ and $g(\cdot)$, and the fact that $W_c(x)$ is a $\mathscr{C}^1$ function that satisfies Eq. 6.13c, there exist positive real numbers k_1 and k_2, such that $|(L_f W_c(x(t)) - L_f W_c(x(t_k))| \leq k_1 |x(t) - x(t_k)|$, $|(L_g W_c(x(t)) - L_g W_c(x(t_k)))u(t))| \leq k_2 |x(t) - x(t_k)|$. Since $f(x)$ and $g(x)$ are continuous functions, and $\mathbb{Z}$ is bounded, there exists a positive real number k_4 and a sampling period Δ', such that $|x(t) - x(t_k)| \leq k_4 \Delta'$ for all $t \in [t_k, t_k + \Delta')$. Also, by the definition of ϕ_{uc}, it follows that $\dot{W}_c(x(t_k)) < -a|W_c(x) - W_c(0)| < -a\rho_m$ holds for all $x \in \mathbb{Z}$, where $\rho_m := \min_{x \in \mathbb{Z}} |W_c(x) - W_c(0)|$. Let $\Delta' < \frac{a\rho_m - \varepsilon}{k_4(k_1+k_2)}$ and $0 \leq \varepsilon < a\rho_m$, where $a > 0$ is used to characterize the set ϕ_{uc}, and substitute the above inequalities obtained from Lipschitz conditions into Eq. 6.22, then it follows that

$$\begin{aligned}\dot{W}_c(x(t),u(t)) &\leq \dot{W}_c(x(t_k),u(t_k)) + k_4(k_1+k_2)\Delta' \\ &< -a\rho_m + k_4(k_1+k_2)\Delta' \\ &< -\varepsilon\end{aligned} \tag{6.23}$$

Eq. 6.23 implies that $W_c(x(t)) < W_c(x(t_k)) \leq \rho_c, \forall\, t > t_k$ and within finite steps, the closed-loop state trajectory $x(t)$ will enter $\mathscr{U}_{\rho_s}$. Hence, $x(t)$ is shown to be bounded in $\mathscr{U}_{\rho_c}$, for all $t \in [t_k, t_k + \Delta')$.

Additionally, consider $x(t_k) \in \mathscr{B}_\delta(x_e)$ where x_e are designed to be saddle points. Since we assume that there exists a set of control actions $\bar{u}(x)$ that decreases $W_c(x)$, $x(t_{k+1})$ is able to move to a smaller level set of $W_c(x)$ and within finite sampling steps leaves $\mathscr{B}_\delta(x_e)$. Moreover, $x(t)$ never returns to $\mathscr{B}_\delta(x_e)$ once it leaves since Eq. 6.23 (i.e., $W_c(x(t)) < W_c(x(t_k)), \forall\, t > t_k$) holds thereafter.

It remains to show that given $x(t_k) \in \mathscr{U}_{\rho_s}$, the trajectory of $x(t)$ will stay in $\mathscr{U}_{\rho_{min}}, \forall\, t \in [t_k, t_k +$

Δ''). Consider Δ'' such that

$$\rho_{min} = \max_{\Delta t \in [0, \Delta'')} \{W_c(x(t_k + \Delta t)) \mid x(t_k) \in \mathscr{U}_{\rho_s},\ u \in U\}. \tag{6.24}$$

Again, there exists a sufficiently small Δ'' such that Eq. 6.24 holds. Therefore, let $\Delta^* = \min\{\Delta', \Delta''\}$, and now we are able to show that for any state $x(t_k) \in \mathscr{U}_{\rho_c}$, $x(t)$ will move towards $\mathscr{U}_{\rho_{min}}$ and remain in $\mathscr{U}_{\rho_c}$ during a sampling period $t \in [t_k,\ t_{k+1})$, where $t_{k+1} := t_k + \Delta$ and $\Delta \in (0, \Delta^*]$. An example of the closed-loop trajectory under the sample-and-hold implementation of $u = \Phi(x)$ and the relationship among the sets $\mathscr{U}_{\rho_c}$, $\mathscr{U}_{\rho_{min}}$ and $\mathscr{U}_{\rho_s}$ are shown in Fig. 6.4. □

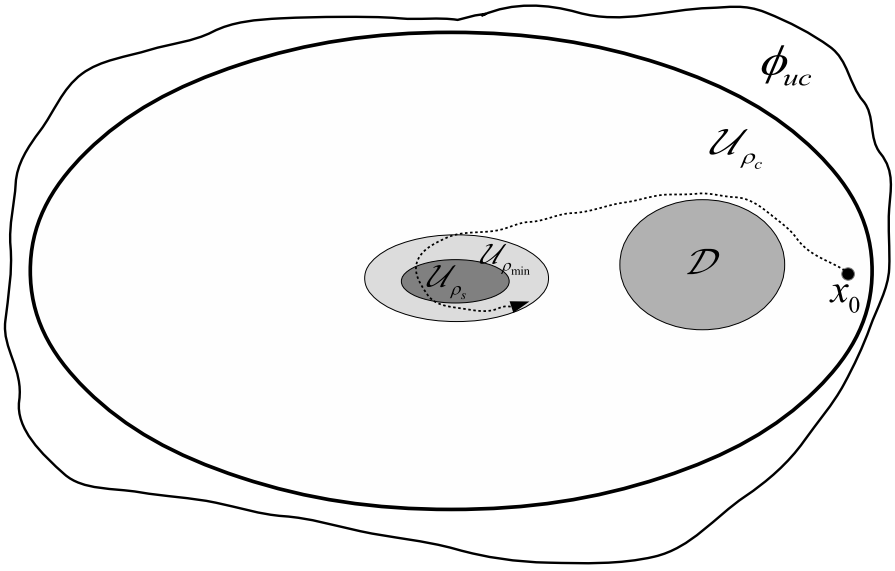


Figure 6.4: A schematic representing the sets $\mathscr{U}_{\rho_c}$, $\mathscr{U}_{\rho_{min}}$ and $\mathscr{U}_{\rho_s}$, where an example of the closed-loop trajectory that originates from $x_0 \in \mathscr{U}_{\rho_c}$ (dotted) is shown to avoid the unsafe region $\mathscr{D}$, and ultimately enter and remain in $\mathscr{U}_{\rho_{min}}$ under the sample-and-hold implementation of $u = \Phi(x) \in U$.

Remark 6.3. *The above proof is based on the assumption that the system of Eq. 6.1 is undisturbed, i.e., $w(t) \equiv 0$. However, when taking the bounded disturbance $|w(t)| \leq \theta$ into account and the CLBF-based controller is applied in a sample-and-hold fashion, we can show that Proposition 6.2 still holds for the system of Eq. 6.1 subject to the bounded disturbance. Specifically, we first derive a similar result for $L_h W_c(x)$ via the local Lipschitz property of $h(\cdot)$: $\exists\ k_3 > 0$, s.t. $|(L_h W_c(x(t)) -$*

$L_h W_c(x(t_k))| \leq k_3|x(t)-x(t_k)|$. *Following that, we obtain similar results for* $\dot{W}_c(x(t),u(t))$ *and* ρ'_{min} *that account for* $w(t)$ *as follows:*

$$\begin{aligned}\dot{W}_c(x(t),u(t)) &\leq \dot{W}_c(x(t_k),u(t_k)) + k_4(k_1+k_2+k_3\theta)\Delta' \\ &< -a\rho_m + k_4(k_1+k_2+k_3\theta)\Delta' \\ &< -\varepsilon\end{aligned} \tag{6.25}$$

$$\rho'_{min} = \max_{\Delta t\in[0,\Delta'')} \{W_c(x(t_k+\Delta t),u,w) \mid x(t_k)\in \mathscr{U}_{\rho_s},\ u\in U,\ |w|\leq\theta\}. \tag{6.26}$$

where $\Delta' < \frac{a\rho_m-\varepsilon}{k_4(k_1+k_2+k_3\theta)}$ *and* $0\leq\varepsilon<a\rho_m$*, respectively. Therefore, by choosing appropriate* Δ' *and* ε *for the sufficiently small bounded disturbance (i.e.,* θ *is sufficiently small),* $\dot{W}_c$ *still remains negative during each sampling period in the presence of disturbance. Additionally, if* $x(t_k)\in\mathscr{B}_\delta(x_e)$*, we again assume that there exists a set of feasible control actions* $\bar{u}(x)$ *that satisfies* $W_c(x(t_{k+1}))<W_c(x(t_k))$*,* $\forall\ |w|\leq\theta$*. On the other hand, based on the definition of* ρ'_{min} *of Eq. 6.26, it is trivial to show that for any* $x(t_k)\in\mathscr{U}_{\rho_s}$*, the trajectory of* $x(t)$ *is guaranteed to stay in* $\mathscr{U}_{\rho'_{min}}, \forall\ t\in[t_k,t_k+\Delta'')$*. The above proof implies that the CLBF-based controller* $u=\Phi(x)\in U$ *in a sample-and-hold fashion is robust to the sufficiently small bounded disturbance.*

Remark 6.4. *We assume that there exists a set of feasible solutions* $\bar{u}(x)\in U$ *in* $\mathscr{B}_\delta(x_e)$ *such that the closed-loop state leaves* $\mathscr{B}_\delta(x_e)$ *in the direction of decreasing* $W_c(x)$*. For example,* $\bar{u}(x)$ *can be determined as* $\bar{u}(x(t_k)) = \arg\min_{u\in U}\{W_c(x(t_{k+1})) \mid\ W_c(x(t_{k+1}))<W_c(x(t_k))\}$*. However, in the absence of input constraints, the fact that* x_e *is a saddle point ensures that there is always a control action (maybe large) that would make* $W_c(x)$ *decrease. Once the state leaves* $\mathscr{B}_\delta(x_e)$ *in the direction of decreasing* $W_c(x)$*, it continues to move towards the origin under* $u=\Phi(x)$*.*

6.5.2 Formulation of CLBF-MPC

The CLBF-MPC design is represented by the following optimization problem [169]:

$$\min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} l_t(\tilde{x}(t), u(t))dt \tag{6.27a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + g(\tilde{x}(t))u(t) \tag{6.27b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{6.27c}$$

$$u(t) \in U, \ \forall\, t \in [t_k, t_{k+N}) \tag{6.27d}$$

$$\dot{W}_c(x(t_k), u(t_k)) \leq \dot{W}_c(x(t_k), \Phi(x(t_k))), \quad \text{if } W_c(x(t_k)) > \rho'_{min} \text{ and } x(t_k) \notin \mathscr{B}_\delta(x_e) \tag{6.27e}$$

$$W_c(\tilde{x}(t)) \leq \rho'_{min}, \ \forall\, t \in [t_k, t_{k+N}), \ \text{ if } W_c(x(t_k)) \leq \rho'_{min} \tag{6.27f}$$

$$W_c(\tilde{x}(t)) < W_c(x(t_k)), \ \forall\, t \in (t_k, t_{k+N}), \ \text{ if } x(t_k) \in \mathscr{B}_\delta(x_e) \tag{6.27g}$$

where $\tilde{x}(t)$ is the predicted state trajectory, $S(\Delta)$ is the set of piecewise constant functions with period Δ, and N is the number of sampling periods in the prediction horizon. $\dot{W}_c(x,u)$ is used to represent $\frac{\partial W_c(x)}{\partial x}(f(x)+g(x)u)$. The cost function $l_t(\tilde{x}(t),u(t))$ satisfies $l_t(0,0)=0$ and $l_t(\tilde{x}(t),u(t)) > 0$, $\forall(\tilde{x}(t),u(t)) \neq (0,0)$ such that the minimum value of the cost function will be attained at the equilibrium point of the system of Eq. 6.1. Let $u^*(t)$ be the optimal solution of the optimization problem of Eq. 6.27 over the prediction horizon $t \in [t_k, t_{k+N})$. We assume that the states of the closed-loop system are measured at each sampling time. Specifically, the above optimization problem is solved based on the measured state $x(t_k)$ at $t = t_k$. After $u^*(t)$, where $t \in [t_k, t_{k+N})$, is obtained from the CLBF-MPC optimization problem, only the first control action of $u^*(t)$ is sent to the control actuators to be applied over the next sampling period. Then, at the next instance of time $t_{k+1} := t_k + \Delta$, the optimization problem is solved again, and the horizon will be rolled one sampling period.

In the optimization problem of Eq. 6.27, the objective function of Eq. 6.27a that is minimized is the integral of $l_t(\tilde{x}(t),u(t))$ over the prediction horizon, where the function $l_t(x,u)$ is not restricted

to a traditional quadratic function. The constraint of Eq. 6.27b is the nominal system of Eq. 6.1 (i.e., $w(t) \equiv 0$) and is used to predict the evolution of the closed-loop state. Eq. 6.27c defines the initial condition of the nominal process system of Eq. 6.27b. Eq. 6.27d defines the input constraints for all the inputs over the entire prediction horizon. The constraint of Eq. 6.27e forces $W_c(\tilde{x})$ along the predicted state trajectories to decrease at least at the rate under $u = \Phi(x)$ when $W_c(x(t_k)) > \rho'_{min}$ and $x(t_k) \notin \mathscr{B}_\delta(x_e)$, while the constraint of Eq. 6.27f activates if $W_c(x(t_k)) \leq \rho'_{min}$ (i.e., $x(t_k)$ enters a small ball around the origin $\mathscr{B}_d(0) := \{x \in \mathbf{R}^n \mid |x| \leq d\}$) so that the states of the closed-loop system will remain inside $\mathscr{B}_d(0)$ afterwards. Additionally, if $x(t_k) \in \mathscr{B}_\delta(x_e)$, the constraint of Eq. 6.27g is activated to decrease $W_c(x)$. Once the state leaves $\mathscr{B}_\delta(x_e)$, it is guaranteed that the state does not return to $\mathscr{B}_\delta(x_e)$ because the state will be driven to smaller level sets of $W_c(x)$ under the constraint of Eq. 6.27e thereafter.

Theorem 6.4 below shows that the control actions computed by the CLBF-MPC of Eq. 6.27 guarantee that the state of the closed-loop system of Eq. 6.1 is always bounded in $\mathscr{U}_{\rho_c}$, and is ultimately bounded in a small region around the origin. In addition, the optimization problems are recursively feasible.

Theorem 6.4. *Consider the system of Eq. 6.1 with a constrained CLBF W_c which has a minimum at the origin, and the set of initial conditions $\mathscr{U}_{\rho_c}$. Given any initial state $x_0 \in \mathscr{U}_{\rho_c}$, it is guaranteed that the optimization problem is feasible for all times under the CLBF-MPC scheme of Eq. 6.27 with sampling period $\Delta \in (0, \Delta^*]$, which is defined in Proposition 6.2. Additionally, for $x_0 \in \mathscr{U}_{\rho_c}$, it is guaranteed that $x(t) \in \mathscr{U}_{\rho_c}$, $\forall\, t \geq 0$, and $\limsup_{t\to\infty} |x(t)| \leq d$.*

Proof. The proof of this proposition consists of two parts. In the first part, we show that for all $x_0 \in \mathscr{U}_{\rho_c}$, the optimization problem of Eq. 6.27 is recursively feasible throughout the entire operating period. Then, we show that under the CLBF-MPC, the trajectory of $x(t)$ is always bounded in $\mathscr{U}_{\rho_c}$, and is ultimately bounded in a small region around the origin $\mathscr{U}_{\rho'_{min}}$.

Part 1 : Assuming that $x(t_k) \in \mathscr{U}_{\rho_c} \backslash \mathscr{U}_{\rho'_{min}}$, $t_k \geq 0$ where ρ'_{min} is defined in Eq. 6.26, the sample-and-hold control law $u(t) = \Phi(x(t_k + i\Delta))$, $i = 0, 1, \ldots, N-1$, and $u(t) = \bar{u}(x)$ are feasible solutions to the optimization problem of Eq. 6.27. Specifically, when $x(t_k) \in \mathscr{U}_{\rho_c} \backslash (\mathscr{U}_{\rho'_{min}} \cup$

$\mathscr{B}_\delta(x_e))$, $u(t) = \Phi(x(t_k + i\Delta))$ satisfies both the input constraint of Eq. 6.27d and the constraint of Eq. 6.27e when the controller is applied in the sample-and-hold fashion. However, if $x(t_k) \in \mathscr{B}_\delta(x_e)$, $u(t) = \bar{u}(x) \in U$ is a set of feasible solutions that satisfies the constraints of Eq. 6.27d and of Eq. 6.27g. The constraint of Eq. 6.27f is not activated in this case.

When $x(t_k) \in \mathscr{U}_{\rho'_{min}}$, $u(t) = \Phi(x(t_k + i\Delta))$, $i = 0, 1, \ldots, N-1$ is again a feasible solution that satisfies the constraints of Eqs. 6.27d, 6.27f. Specifically, if $x(t_k) \in \mathscr{U}_{\rho_s} \subset \mathscr{U}_{\rho'_{min}}$, it is guaranteed that the constraint of Eq. 6.27f is satisfied according to the definition of ρ'_{min} of Eq. 6.26. However, if $x(t_k) \in \mathscr{U}_{\rho'_{min}} \backslash \mathscr{U}_{\rho_s}$, based on the proof in Proposition 6.2, it follows that the sample-and-hold controller $u(t) = \Phi(x(t_k + i\Delta))$ guarantees $\dot{W}_c(x) < -\varepsilon$ over a sampling period, which implies that $W_c(x(t_{k+1})) \leq W_c(x(t_k)) \leq \rho'_{min}$. Therefore, at every sampling time, if $x(t_k) \in \mathscr{U}_{\rho_c}$, a feasible solution to the optimization problem of Eq. 6.27 exists.

Part 2 : We now prove that if $x_0 \in \mathscr{U}_{\rho_c}$, it holds that $x(t) \in \mathscr{U}_{\rho_c}$, $\forall\, t \geq 0$. We first consider the case of a bounded unsafe region $\mathscr{D}_b$. Since the initial condition x_0 is in the set $\mathscr{U}_{\rho_c}$, it follows that under the constraints of Eqs. 6.27d-6.27g, $x(t) \in \mathscr{U}_{\rho_c}$, $\forall\, t \geq 0$ by letting $t_k = 0$ for the result of $W_c(x(t)) < W_c(x(t_k)) \leq \rho_c, \forall\, t > t_k$ from Proposition 6.2. Therefore, the assumption that $x(t_k) \in \mathscr{U}_{\rho_c}$ at $t = t_k, t_k \geq 0$ in Part 1 is also proved.

Finally, let $x_0 \in \mathscr{U}_{\rho_c} \backslash \mathscr{U}_{\rho'_{min}}$, we will show that $x(t)$ ultimately enters $\mathscr{U}_{\rho'_{min}}$ and remains there for all subsequent times. It follows that $W_c(x(t+\Delta)) < W_c(x(t))$ holds when $x(t) \in \mathscr{U}_{\rho_c} \backslash (\mathscr{U}_{\rho'_{min}} \cup \mathscr{B}_\delta(x_e))$ and $x(t) \in \mathscr{B}_\delta(x_e)$ from the proof in Proposition 6.2. This implies that within finite time t_s, the trajectory will enter $\mathscr{U}_{\rho'_{min}}$. Additionally, it has been shown in Part 1 that if $x(t) \in \mathscr{U}_{\rho'_{min}}$, the constraint of Eq. 6.27f is satisfied according to the definition of $\mathscr{U}_{\rho'_{min}}$, and hence there always exists a set of control actions such that $W_c(x(t)) \leq \rho'_{min}$, $\forall\, t \geq t_s$. Note that $W_c(\cdot)$ is a continuous function of the state, and thus, given the real number ρ'_{min}, one can find a positive real number d, such that $W_c(x(t)) \leq \rho'_{min}$ implies $\limsup_{t\to\infty} |x(t)| \leq d$.

On the other hand, if the unsafe region is an unbounded set $\mathscr{D}_u$ in state-space, if follows that the origin is the only stationary point in state-space (i.e., $\mathbb{X}_e = \emptyset$) and the constraint of Eq. 6.27g will never be activated. Therefore, under the constraint of Eq. 6.27e, $\dot{W}_c$ is rendered negative and

the state of the closed-loop system of Eq. 6.1 is driven towards $\mathscr{U}_{\rho'_{min}}$. Finally, the closed-loop state is bounded in $\mathscr{U}_{\rho'_{min}}$ under the constraint of Eq. 6.27f. □

Remark 6.5. *We note that Theorem 6.4 applies to both the nominal closed-loop system of Eq. 6.1, i.e., $\dot{x} = f(x) + g(x)u + h(x)w$ with $w(t) \equiv 0$, and the closed-loop system of Eq. 6.1 subject to bounded disturbances (i.e., $|w| \leq \theta$) under the sample-and-hold implementation of CLBF-MPC. The case of nominal closed-loop system is straightforward since the nominal system of Eq. 6.1 is used as the prediction process model in the formulation of CLBF-MPC of Eq. 6.27, which implies that the actual closed-loop states are consistent with the predicted states under the CLBF-MPC, and therefore, closed-loop stability and safety are guaranteed following the above proof. However, for the system subject to bounded disturbances, it is shown in Proposition 6.2 that for sufficiently small disturbances $|w(t)| \leq \theta$ and sufficiently small sampling period Δ, $\dot{W}_c$ of the uncertain closed-loop system of Eq. 6.1 still satisfies $\dot{W}_c < -\varepsilon$ for all $x \in \mathscr{U}_{\rho_c} \backslash (\mathscr{U}_{\rho'_{min}} \cup \mathscr{B}_\delta(x_e))$. Additionally, if $x \in \mathscr{U}_{\rho'_{min}}$ or $x \in \mathscr{B}_\delta(x_e)$, the constraints of Eq. 6.27f and of Eq. 6.27g still hold since ρ'_{min} of Eq. 6.26 and $\bar{u}(x)$ are determined accounting for the impact of the bounded disturbances. Therefore, all the constraints of CLBF-MPC are satisfied and Theorem 6.4 holds for the uncertain closed-loop system of Eq. 6.1 with $|w| \leq \theta$.*

Remark 6.6. *It should be noted that the problem of convergence to x_e instead of the origin can be solved by taking advantage of the CLBF-MPC. The constraint of Eq. 6.27g requires $W_c(x)$ to decrease if $x(t_k) \in \mathscr{B}_\delta(x_e)$, which drives the state out of $\mathscr{B}_\delta(x_e)$ in the direction of decreasing $W_c(x)$. Additionally, since in general, the objective function of the CLBF-MPC of Eq. 6.27a penalizes the distances between states and the origin and also control actions, the objective function value becomes large if the state converges to any points other than the origin (e.g., x_e). Therefore, CLBF-MPC will try to avoid converging to x_e by optimizing control actions in a sample-and-hold fashion (i.e., discontinuous control actions) and taking future cost values into account.*

6.5.3 Application to a Chemical Process Example

In this section, we utilize a chemical process example to illustrate the application of the proposed CLBF-MPC method. Consider a well-mixed, non-isothermal continuous stirred tank reactor (CSTR) where an irreversible first-order exothermic reaction takes place. The reaction converts the reactant A to the product B via the chemical reaction $A \rightarrow B$. A heating jacket that supplies or removes heat from the reactor is used. The CSTR dynamic model derived from material and energy balances is given below:

$$\frac{dC_A}{dt} = \frac{F}{V_L}(C_{A0} - C_A) - k_0 e^{-E/RT} C_A + w_1 \tag{6.28a}$$

$$\frac{dT}{dt} = \frac{F}{V_L}(T_0 - T) - \frac{\Delta H k_0}{\rho_L C_p} e^{-E/RT} C_A + \frac{Q}{\rho_L C_p V_L} + w_2 \tag{6.28b}$$

where C_A is the concentration of reactant A in the reactor, T is the temperature of the reactor, Q denotes the heat supply/removal rate, and V_L is the volume of the reacting liquid in the reactor. The feed to the reactor contains the reactant A at a concentration C_{A0}, temperature T_0, and volumetric flow rate F. The liquid has a constant density of ρ_L and a heat capacity of C_p. k_0, E and ΔH are the reaction pre-exponential factor, activation energy and the enthalpy of the reaction, respectively. Process parameter values are listed in Table 6.1. The control objective is to operate the CSTR at the equilibrium point $(C_{As},\ T_s) = (0.57\ kmol/m^3,\ 395.3\ K)$ and maintain the state in a safe region of state-space by manipulating the heat input rate $\Delta Q = Q - Q_s$, and the inlet concentration of species A, $\Delta C_{A0} = C_{A0} - C_{A0_s}$. The input constraints for ΔQ and ΔC_{A0} are $|\Delta Q| \leq 0.0167\ kJ/min$ and $|\Delta C_{A0}| \leq 1\ kmol/m^3$, respectively.

To place Eq. 6.28 in the form of nonlinear systems of Eq. 6.1, deviation variables are used in this example, such that the equilibrium point of the system is at the origin of the state-space. $x^T = [C_A - C_{As}\ \ T - T_s]$ represents the state vector in deviation variable form, $u^T = [\Delta C_{A0}\ \ \Delta Q]$ represents the manipulated input vector in deviation variable form, and $w^T = [w_1\ \ w_2]$ is the bounded disturbance vector of Gaussian distribution with zero mean and standard deviation $\sigma_1 = 1.0\ kmol/(m^3\ min)$, $\sigma_2 = 3.5\ K/min$. The upper bound for disturbances $|w_1| \leq 1.0\ kmol/(m^3\ min)$

Table 6.1: Parameter values of the CSTR with a first-order reaction.

$T_0 = 310\ K$	$F = 100 \times 10^{-3}\ m^3/min$
$V_L = 0.1\ m^3$	$E = 8.314 \times 10^4\ kJ/kmol$
$k_0 = 72 \times 10^9\ min^{-1}$	$\Delta H = -4.78 \times 10^4\ kJ/kmol$
$C_p = 0.239\ kJ/(kg\ K)$	$R = 8.314\ kJ/(kmol\ K)$
$\rho_L = 1000\ kg/m^3$	$C_{A0_s} = 1.0\ kmol/m^3$
$Q_s = 0.0\ kJ/min$	$C_{A_s} = 0.57\ kmol/m^3$
$T_s = 395.3\ K$	

and $|w_2| \leq 3.17\ K/min$ are approximated via simulation runs under various sizes of disturbances. The control Lyapunov function is designed using the standard quadratic form $V(x) = x^T P x$ with $P = \begin{bmatrix} 9.35 & 0.41 \\ 0.41 & 0.02 \end{bmatrix}$.

6.5.3.1 Case study: bounded unsafe region

We first demonstrate the application of CLBF-MPC to a bounded unsafe region $\mathscr{D}_b$ located within the set ϕ_{uc}. The unsafe region is defined as an ellipse: $\mathscr{D}_b := \{x \in \mathbf{R}^2 \mid F(x) = \frac{(x_1+0.22)^2}{1} + \frac{(x_2-4.6)^2}{1\times 10^4} < 2 \times 10^{-4}\}$. H is defined as $H := \{x \in \mathbf{R}^2 \mid F(x) < 4 \times 10^{-4}\}$ such that it satisfies $\mathscr{D}_b \subset H \subset \phi_{uc}$ in Proposition 6.1. The control barrier function $B(x)$ is defined as follows.

$$B(x) = \begin{cases} e^{\frac{\lambda F^2(x)}{F(x)-4\times 10^{-4}}} - e^{-2\lambda \times 10^{-4}}, & \text{if } x \in H \\ -e^{-2\lambda \times 10^{-4}}, & \text{if } x \notin H \end{cases} \tag{6.29}$$

where $\lambda > 0$ is a parameter that can be used to adjust the value of $B(x)$ in characterizing the set ϕ_{uc}. From Eq. 6.29, it is guaranteed that $B(x)$ is positive in the unsafe region $\mathscr{D}_b$. Then, the control Lyapunov-barrier function $W_c(x) = V(x) + \mu B(x) + \nu$ is constructed following the rules in Proposition 6.1, where the parameters are determined as follows, $\lambda = 0.001$, $\rho_c = 0$, $c_1 = 0.001$, $c_2 = 10$, $c_3 = \max_{x\in\partial H} |x|^2 = 34.8$, $c_4 = \min_{x\in\partial \mathscr{D}_b} |x|^2 = 16.85$, and $\nu = \rho_c - c_1 c_4 = -1.685 \times 10^{-2}$. Hence, μ is chosen to be 5000 to satisfy Eq. 6.19. Based on the above $W_c(x)$, x_e is calculated

to be a saddle point (-0.235, 4.83) in state-space.

The objective function of the CLBF-MPC in this example is to drive the system to its equilibrium point while minimizing the heat supply/removal rate, and the feed reactant concentration as well, and is given as follows,

$$l_t(\tilde{x}, u) = |\tilde{x}(t)|^2_{Q_L} + |u(t)|^2_{R_L} \tag{6.30}$$

where the weighting matrices for the states and inputs are chosen to be $Q_L = \begin{bmatrix} 1000 & 0 \\ 0 & 10 \end{bmatrix}$ and $R_L = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}$, respectively, such that the term related to the states and the term related to the inputs are on the same order of magnitude in Eq. 6.30 to penalize both state and input deviations from the steady-state significantly. In the simulations below, the process model of Eq. 6.28 is integrated numerically using the explicit Euler method with an integration time step of $h_c = 10^{-5}$ *min*. The MPC sampling period and the prediction horizon were chosen to be $\Delta = 2 \times 10^{-3}$ *min* and $N = 10$, under which the desired closed-loop performance is achieved with high computational efficiency (i.e., the control action calculation is done within the sampling period). The constrained nonlinear optimization problem is solved using the IPOPT software package ([158]) with a 4-core CPU desktop.

Scenario 1: We first carry out the closed-loop simulation for the nominal CSTR system (i.e., no disturbances w) in the presence of a bounded unsafe region $\mathscr{D}$. We chose the subset $\mathscr{U}_\rho \subset \mathscr{U}_{\rho_c}$ as the safe operating region and set an initial condition that is far away from the set $\mathscr{D}$. Starting from the initial condition (x_1, x_2)=(0.2, -5), it is demonstrated that the stabilization of the closed-loop system can be achieved (the green trajectory in Fig. 6.5), and the states always remain in $\mathscr{U}_\rho$. Additionally, another three initial conditions (-0.19, 5.5), (-0.35, 7) and (-0.235, 6.5) are chosen to start the system from where the state encounters the unsafe region $\mathscr{D}$ on its way to the origin under the CLBF-MPC as shown in Fig. 6.5. All three demonstrate that the states avoid the unsafe region $\mathscr{D}$ and ultimately converge to the origin.

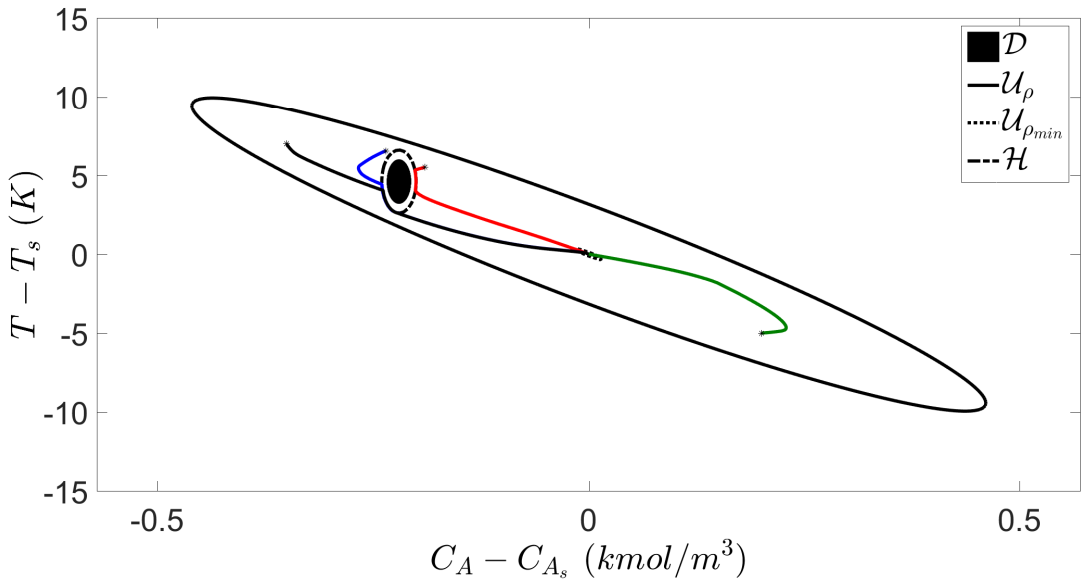


Figure 6.5: Closed-loop state trajectories under CLBF-MPC for four different initial conditions (0.2, -5) (green), (-0.19, 5.5) (red), (-0.35, 7) (black) and (-0.235, 6.5) (blue). The set of unsafe states $\mathscr{D}$ is shaded in solid black area and the set $\mathscr{U}_\rho$ is the region between the largest ellipse and the set H.

Scenario 2: We now compare the closed-loop performance of the CSTR system under CLBF-MPC with that under a non-Lyapunov-based MPC with a state constraint to avoid $\mathscr{D}$ and a terminal constraint to guarantee closed-loop stability that is described by the following optimization problem.

$$\min_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} l_t(\tilde{x}(t), u(t))dt \tag{6.31a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + g(\tilde{x}(t))u(t) \tag{6.31b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{6.31c}$$

$$u(t) \in U, \ \forall\, t \in [t_k, t_{k+N}) \tag{6.31d}$$

$$\tilde{x}(t) \in \mathscr{U}_\rho, \ \forall\, t \in [t_k, t_{k+N}), \ \text{ if } x(t_k) \in \mathscr{U}_\rho \tag{6.31e}$$

$$\tilde{x}(t_{k+N}) \in \mathscr{U}_{\rho_{min}} \tag{6.31f}$$

It is demonstrated in Fig. 6.6 that starting from the same initial condition (-0.235, 6.5), the

state under CLBF-MPC (black solid) will first reach the boundary of H, then avoid the unsafe region $\mathscr{D}$ by passing around it, and finally, move towards the origin. However, under MPC with state constraints, it is demonstrated that the optimization problem becomes infeasible when the trajectory gets close to the boundary of the unsafe region. In this case, we deactivate the state constraint and apply the feasible solution of the optimization problem of MPC with terminal constraint only such that the trajectory crosses the unsafe region but can still move towards the origin. Therefore, CLBF-MPC outperforms the standard MPC with state constraints since it reconciles the tasks of safety and closed-loop stability with guaranteed recursive feasibility.

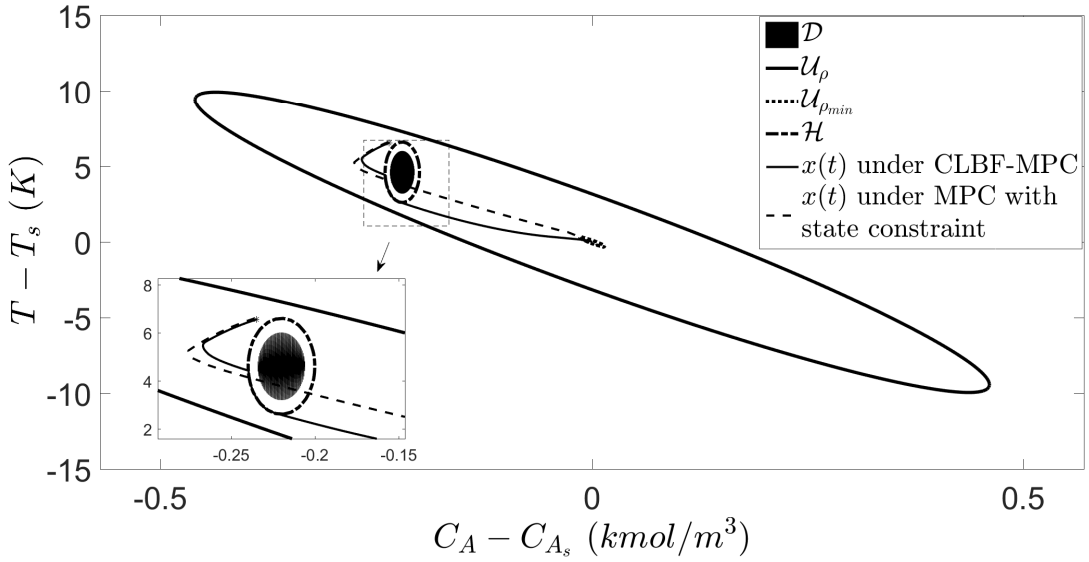


Figure 6.6: Closed-loop state profiles under the CLBF-MPC of Eq. 6.27 (solid) and under the MPC with state constraints (dashed), where the unsafe region $\mathscr{D}$ is an obstacle for the closed-loop state trajectory starting from the initial condition (-0.235, 6.5).

Scenario 3: We carry out the closed-loop simulation subject to bounded disturbance. It is demonstrated in Fig. 6.7 that the CLBF-MPC can still guarantee safety and closed-loop stability. The corresponding input profiles are also shown in Fig 6.8, in which it is seen that the control actions oscillate around the steady state due to the disturbance.

Scenario 4: Lastly, to demonstrate the advantages of the proposed CLBF-MPC control scheme compared to the case of using explicit CLBF-based control law of Eq. 6.11 all the time, the simulation results of the closed-loop state and inputs profiles for the same initial condition (-0.235,

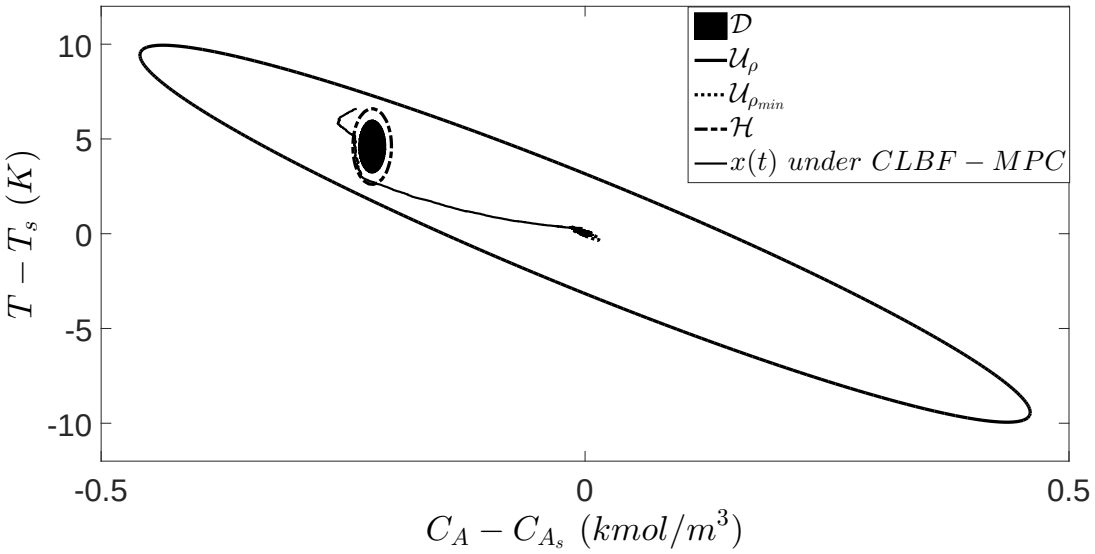


Figure 6.7: Closed-loop state profile for the initial condition (-0.235, 6.5) under the CLBF-MPC of Eq. 6.27 (solid) subject to bounded disturbance.

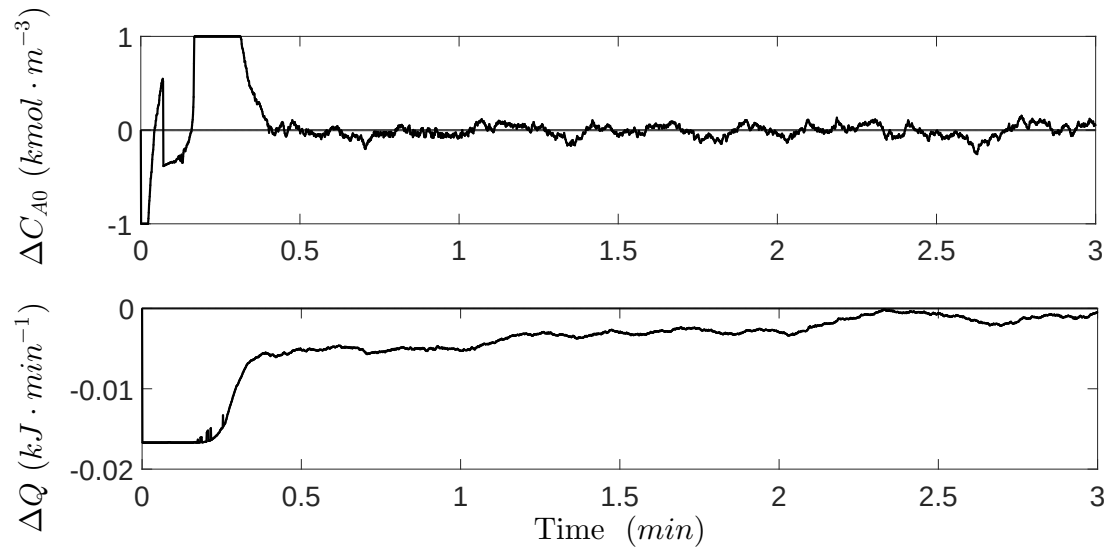


Figure 6.8: Manipulated input profiles ($u_1 = \Delta C_{A0}$ and $u_2 = \Delta Q$) for the initial condition (-0.235, 6.5) under the CLBF-MPC of Eq. 6.27 subject to bounded disturbance.

6.5) are shown in Fig. 6.9 and Fig. 6.10, respectively. In Fig. 6.10, it is observed that under the CLBF-based control law of Eq. 6.11, the inlet concentration of the reactant and the heat input rate start oscillating heavily from $t = 0.003$ *min* to $t = 0.2$ *min*, and correspondingly, the oscillation arises in the state trajectory near the boundary of H. The reason for the oscillation is that the intrinsic dynamics of the closed-loop system force the states to go towards and cross the unsafe region, yet the constraints of CLBF-MPC prevent this undesirable behavior due to the dramatic increase in the values of W_c inside the unsafe region (barrier function dominates). By balancing these two opposite effects, the control action becomes oscillating when the state moves around the boundary of H. Additionally, under the proposed CLBF-MPC control scheme, the dynamic performance was improved since MPC has the ability to anticipate future state behavior and can take control actions accordingly.

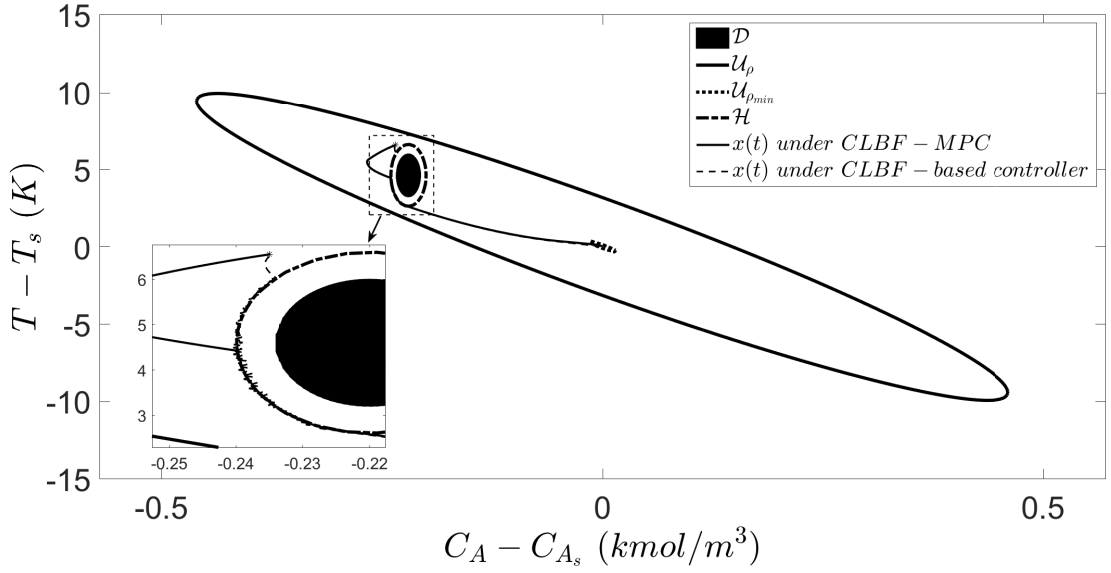


Figure 6.9: Closed-loop state profiles for the initial condition (-0.235, 6.5) under the CLBF-MPC of Eq. 6.27 (solid) and under the CLBF-based controller of Eq. 6.11 (dashed).

Furthermore, it is calculated based on the simulation results under CLBF-MPC that the total consumptions of reactant ΔC_{A0} and of energy ΔQ within the operating time $t_s = 3$ *min* are 0.268 $kmol/m^3$ and 0.006 *kJ*, respectively, which represent improvements of 13% and 25%, respectively, compared to 0.308 $kmol/m^3$ and 0.008 *kJ* under the explicit CLBF-based controller.

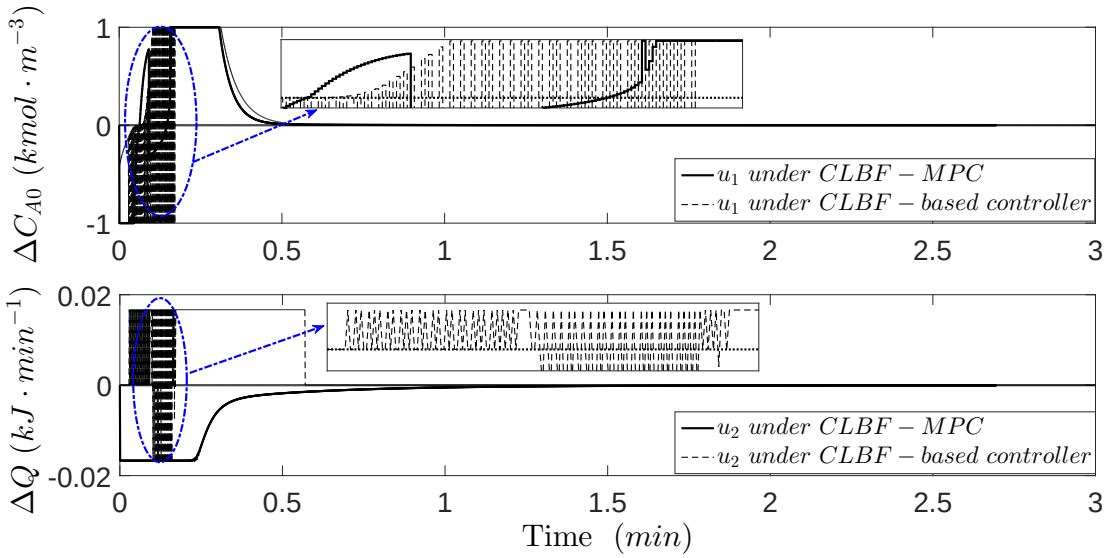


Figure 6.10: Manipulated input profiles ($u_1 = \Delta C_{A0}$ and $u_2 = \Delta Q$) for the initial condition (-0.235, 6.5) under the CLBF-MPC of Eq. 6.27 (solid) and under the CLBF-based controller of Eq. 6.11 (dashed).

Therefore, in this case, the CLBF-MPC of Eq. 6.27 outperforms the explicit CLBF-based controller of Eq. 6.11 due to the smoother control actions and reduced control energy consumptions.

6.5.3.2 Case study: unbounded unsafe region

The closed-loop simulation for the CSTR system with an unbounded unsafe region is carried out in this section. Specifically, the unsafe region is defined as an unbounded set with high temperature and concentration: $\mathscr{D}_u := \{x \in \mathbf{R}^2 \mid F(x) = x_1 + x_2 > 7.2\}$. H is defined as $H := \{x \in \mathbf{R}^2 \mid F(x) > 6.8\}$. The control barrier function $B(x)$ is defined as follows.

$$B(x) = \begin{cases} e^{F(x)-7.2} - 2 \times e^{-0.4}, & \text{if } x \in H \\ -e^{-0.4}, & \text{if } x \notin H \end{cases} \tag{6.32}$$

The control Lyapunov-barrier function $W_c(x) = V(x) + \mu B(x) + \nu$ is constructed with the following parameters: $\rho_c = 0$, $c_1 = 0.001$, $c_2 = 10$, $c_3 = 98.78$, $c_4 = 51.99$, and $\nu = \rho_c - c_1 c_4 = -1.685 \times 10^{-2}$. Hence, μ is chosen to be 1500 to satisfy Eq. 6.19. For simplicity, we only discuss the scenario of the nominal CSTR system under CLBF-MPC. It is demonstrated in Fig. 6.11 that

under the CLBF-MPC of Eq. 6.27, all the trajectories with initial states inside $\mathscr{U}_\rho$ avoid the unsafe region $\mathscr{D}_u$ on the top and converge to $\mathscr{U}_{\rho_{min}}$.

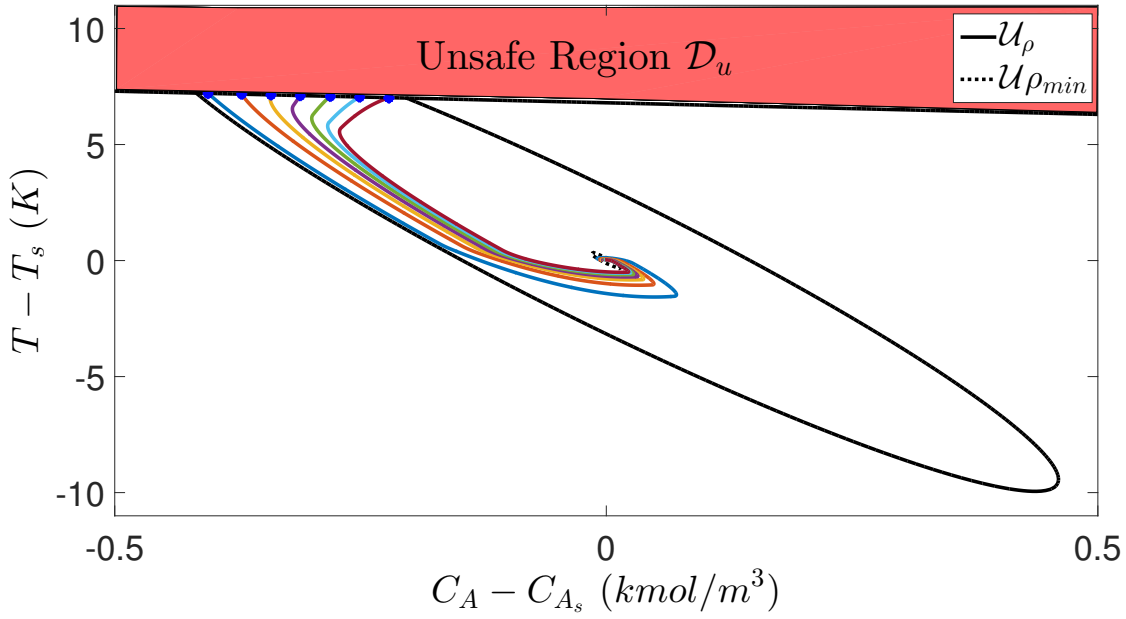


Figure 6.11: Closed-loop state trajectories for the system of Eq. 6.28 under CLBF-MPC with different initial conditions marked by stars. The set of unbounded unsafe states $\mathscr{D}_u$ is the red area on the top.

Therefore, from the above case studies of a bound and an unbounded unsafe region, it is demonstrated that simultaneous closed-loop stability and process operational safety are achieved under the CLBF-MPC of Eq. 6.27 in the sense that for any initial state $x_0 \in \mathscr{U}_\rho \subset \mathscr{U}_{\rho_c}$, the closed-loop state is guaranteed to stay inside $\mathscr{U}_\rho$ and avoid the unsafe region for all times, and will converge to a small neighborhood $\mathscr{U}_{\rho_{min}}$ around the origin ultimately.

6.6 CLBF-based Economic Model Predictive Control

By incorporating CLBF-based constraints into tracking MPC, the state of a closed-loop nonlinear system can be driven to its set point while avoiding a bounded/unbounded unsafe region in state-space. However, given that the steady-state operation may not be optimal for industrial process operation as demonstrated in the previous chapter, economic model predictive control

(EMPC) that optimizes directly in real time the economic performance of the process is an efficient method to improve process economic performance while maintaining stable operation. Therefore, based on the Lyapunov-based EMPC of Eq. 3.1 and the stabilizability and safety assumptions in Section 6.4.1, we present the design of CLBF-based EMPC (CLBF-EMPC) that ensures closed-loop stability, process operational safety, and economic optimality simultaneously. It should be noted that simultaneous closed-loop stability and operational safety now represent the boundedness of the state in a safe operating region only as EMPC does not require the convergence of state to the steady-state.

6.6.1 CLBF-based EMPC formulation

The CLBF-EMPC design is represented by the following optimization problem [175]:

$$\max_{u(t)\in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(\tau), u(\tau))\, d\tau \tag{6.33a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + g(\tilde{x}(t))u(t) \tag{6.33b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{6.33c}$$

$$u(t) \in U,\ \forall\, t \in [t_k, t_{k+N}) \tag{6.33d}$$

$$W_c(\tilde{x}) \leq \rho_e,\ \ \forall\, t \in [t_k, t_{k+N}),\ \ \text{if } x(t_k) \in \mathscr{U}_{\rho_e} \tag{6.33e}$$

$$\dot{W}_c(x(t_k), u(t_k)) \leq \dot{W}_c(x(t_k), \Phi(x(t_k))),\ \ \text{if } x(t_k) \in \mathscr{U}_{\rho} \backslash \mathscr{U}_{\rho_e} \tag{6.33f}$$

where the notation follows that for CLBF-MPC in Eq. 6.27. The optimization problem of Eq. 6.33 optimizes the time integral of the cost function $l_e(x,u)$ of Eq. 6.33a that represents process economic benefits while satisfying the constraints of Eqs. 6.33b-6.33f. Specifically, the nominal process model of Eq. 6.33b is used as the prediction of CLBF-EMPC. Eq. 6.33c defines the initial condition for the optimization problem of Eq. 6.33 using the measurement of the process state at the current time t_k. Eq. 6.33d defines the input constraints applied over the prediction horizon. If $x(t_k)$ is inside $\mathscr{U}_{\rho_e}$, the Mode 1 constraint of Eq. 6.33e is applied to maintain the predicted

closed-loop state within the set $\mathscr{U}_{\rho_e} \subset \mathscr{U}_\rho$, which is designed to make the safe operating region $\mathscr{U}_\rho$ a forward invariant set in the presence of sufficiently small disturbances (i.e., $|w(t)| \leq \theta$) and also include the states $x_e \in \mathbb{X}_e$ inside (i.e., $\mathscr{B}_\delta(x_e) \subset \mathscr{U}_{\rho_e}$). Under the Mode 2 constraint of Eq. 6.33f, the contractive constraint is activated only for the next sampling step to decrease the value of $W_c(x)$, such that the closed-loop state will move back into $\mathscr{U}_{\rho_e}$ within finite sampling steps. The CLBF-EMPC is implemented in a sample-and-hold fashion, which implies only the first step of the optimized input trajectory will be applied over the next sampling period.

Before we demonstrate closed-loop stability and safety under CLBF-EMPC in Theorem 6.5, we first establish a few propositions that will be used in the proof of theorem. Specifically, Proposition 6.3 gives the upper bound on the difference between the evolutions of the trajectories of the nominal system (i.e., $w(t) \equiv 0$) and the disturbed system of Eq. 6.1. Proposition 6.4 establishes the relationship of the disturbance bound, Lipschitz constants, and the sampling period that is required to maintain $\dot{W}_c$ negative during one sampling period, which will be utilized in the proof of closed-loop stability and safety of the CLBF-EMPC in Theorem 6.5. Also, it should be pointed out that for the CLBF-EMPC of Eq. 6.33, we omit the case where $x_0 \in \phi_{uc} \backslash (\mathscr{D} \cup \mathscr{U}_{\rho_c})$ and only consider the initial condition $x_0 \in \mathscr{U}_\rho \subset \mathscr{U}_{\rho_c}$ since closed-loop stability under CLBF-EMPC represents the boundedness of the state $x(t)$ within an invariant set $\mathscr{U}_\rho$.

Proposition 6.3. *Consider the system of Eq. 6.1, i.e., $\dot{x} = F(x,u,w) := f(x) + g(x)u + h(x)w$ and the nominal system $\dot{\hat{x}} = F(\hat{x},u,0)$ (i.e., $w(t) \equiv 0$) with initial conditions $x_0 = \hat{x}_0 \in \mathscr{U}_\rho \subset \mathscr{U}_{\rho_c}$. There exists a class $\mathscr{K}$ function $f_w(\cdot)$ and a positive constant β such that the following inequalities hold $\forall x, \hat{x} \in \mathscr{U}_\rho$ and $w(t) \in W$:*

$$|x(t) - \hat{x}(t)| \leq f_w(t) := \frac{L_w \theta}{L_x}(e^{L_x t} - 1) \tag{6.34a}$$

$$W_c(x) \leq W_c(\hat{x}) + \alpha_4(\alpha_1^{-1}(\rho - \rho_0))|x - \hat{x}| + \beta |x - \hat{x}|^2 \tag{6.34b}$$

Proof. Let the error vector $e(t) = x(t) - \hat{x}(t)$. The derivative of $e(t)$ can be obtained as follows:

$$|\dot{e}(t)| = |F(x(t), u(t), w(t)) - F(\hat{x}(t), u(t), 0)| \tag{6.35}$$

Following Eq. 6.14b, it is obtained that

$$|\dot{e}(t)| \leq L_x|x(t) - \hat{x}(t)| + L_w|w(t)| \leq L_x|e(t)| + L_w|\theta| \tag{6.36}$$

Therefore, for all $x(t), \hat{x}(t) \in \mathscr{U}_\rho, |w(t)| \leq \theta$ and zero initial condition (i.e., $e(0) = 0$), we can derive the upper bound of the norm of the error vector as follows:

$$|e(t)| = |x(t) - \hat{x}(t)| \leq \frac{L_w\theta}{L_x}(e^{L_x t} - 1) \tag{6.37}$$

Subsequently, we prove Eq. 6.34b holds for all $x, \hat{x} \in \mathscr{U}_\rho$ by using the Taylor series expansion of $W_c(x)$ around $\hat{x}$ as follows:

$$W_c(x) \leq W_c(\hat{x}) + \frac{\partial W_c(\hat{x})}{\partial x}|x - \hat{x}| + \beta|x - \hat{x}|^2 \tag{6.38}$$

Substituting Eq. 6.13a and Eq. 6.13c into Eq. 6.38, it follows that

$$W_c(x) \leq W_c(\hat{x}) + \alpha_4(\alpha_1^{-1}(\rho - \rho_0))|x - \hat{x}| + \beta|x - \hat{x}|^2 \tag{6.39}$$

□

Proposition 6.4. *Consider the system of Eq. 6.1 under the controller $u = \Phi(x) \in U$, designed based on W_c with its minimum at the origin and meeting Eq. 6.12 and Eq. 6.13, implemented in sample-and-hold. Let $\varepsilon_w > 0$, $\Delta^* > 0$, $\rho > \rho_e$ satisfy*

$$-\alpha_3(\alpha_2^{-1}(\rho_e - \rho_0)) + L_x^{'}M\Delta^* + L_w^{'}\theta \leq -\varepsilon_w/\Delta^* \tag{6.40}$$

Then, for any $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, the following inequality holds under $u = \Phi(x) \in U$:

$$W_c(x(t)) \leq W_c(x(t_k)),\ \forall t \in [t_k, t_{k+1}) \tag{6.41}$$

Proof. Assuming $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, we prove that within one sampling period, the value of $W_c(x)$ is decreasing under the controller $u(t) = \Phi(x(t_k)) \in U$. The time derivative of the CLBF $W_c(x)$ along the trajectory $x(t)$ of the nominal system of Eq. 6.1 in $t \in [t_k, t_{k+1})$ is given by:

$$\dot{W}_c(x(t)) = \frac{\partial W_c(x(t))}{\partial x} F(x(t), \Phi(x(t_k)), w(t)) \tag{6.42}$$

Adding $\frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), \Phi(x(t_k)), 0)$ to both sides and using Eq. 6.13b, the following inequality is obtained:

$$\begin{aligned} \dot{W}_c(x(t)) \leq & -\alpha_3(|x(t_k)|) + \frac{\partial W_c(x(t))}{\partial x} F(x(t), \Phi(x(t_k)), w(t)) \\ & - \frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), \Phi(x(t_k)), 0) \end{aligned} \tag{6.43}$$

Based on the inequalities of Eq. 6.13a and Eq. 6.14, the upper bound of $\dot{W}_c(x(t))$ is derived as follows for $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$:

$$\begin{aligned} \dot{W}_c(x(t)) &\leq -\alpha_3(\alpha_2^{-1}(\rho_e - \rho_0)) + L_x^{'}|x(t) - x(t_k)| + L_w^{'}\theta \\ &\leq -\alpha_3(\alpha_2^{-1}(\rho_e - \rho_0)) + L_x^{'} M \Delta^* + L_w^{'}\theta \end{aligned} \tag{6.44}$$

Therefore, if Eq. 6.40 is satisfied, $\dot{W}_c(x(t)) \leq -\varepsilon_w/\Delta^*$ holds for all $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, $t \in [t_k, t_{k+1})$. Through the integral of the above equation, we obtain that $W_c(x(t_{k+1})) \leq W_c(x(t_k)) - \varepsilon_w$, and also the conclusion shown in Eq. 6.41. □

Based on the CLBF-EMPC of Eq. 6.33, the following theorem establishes that under the sample-and-hold implementation of the solution of the CLBF-EMPC of Eq. 6.33, both closed-loop stability and process operational safety are guaranteed for the system of Eq. 6.1, and the optimization problem is recursively feasible.

Theorem 6.5. *Consider the system of Eq. 6.1 with a constrained CLBF $W_c(x) : \mathbf{R}^n \rightarrow \mathbf{R}$ that has its minimum at the origin and meets Eqs. 6.12 and 6.13. Let $\Delta \leq \Delta^*$ and $\rho > \rho_e$ satisfy Eq. 6.40 and ρ_e be determined as follows:*

$$\rho_e \leq \rho - \alpha_4(\alpha_1^{-1}(\rho - \rho_0))f_w(\Delta) - \beta(f_w(\Delta))^2 \tag{6.45}$$

Given any initial state $x_0 \in \mathscr{U}_\rho$, it is guaranteed under the CLBF-EMPC of Eq. 6.33, $x(t) \in \mathscr{U}_\rho$, $\forall t \geq 0$ for the closed-loop system of Eq. 6.1, where $\mathscr{U}_\rho \subset \mathscr{U}_{\rho_c}$ and $\mathscr{U}_\rho \cap \mathscr{D} = \emptyset$.

Proof. To prove closed-loop stability and safety of the system of Eq. 6.1 subject to small bounded disturbances (i.e., $|w(t)| \leq \theta$) under CLBF-EMPC, we first prove that under the Mode 1 constraint of Eq. 6.33e of CLBF-EMPC, the closed-loop state is always bounded in the stability and safety region $\mathscr{U}_\rho$ (stability comes from the invariance of the level set of $W_c(x)$ while safety is due to the fact that $\mathscr{U}_\rho \cap \mathscr{D} = \emptyset$). We then prove that if the system operates in the second operation mode (i.e., the Mode 2 constraint of Eq. 6.33f when $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$), the closed-loop state will move towards the origin, and enter $\mathscr{U}_{\rho_e}$ in finite sampling steps. Finally, we prove that the CLBF-EMPC of Eq. 6.33 is solved with recursive feasibility for all states $x(t) \in \mathscr{U}_\rho$.

Part 1 : We prove that if $x(t_k) \in \mathscr{U}_{\rho_e}$, $t_k \geq 0$, the closed-loop state $x(t) \in \mathscr{U}_\rho$, $\forall t \in [t_k, t_{k+1}]$ holds. Since the state $x(t_k)$ at $t = t_k$ is assumed to be in the set $\mathscr{U}_{\rho_e}$, the CLBF-EMPC of Eq. 6.33 operates in the first operation mode (i.e., the Mode 1 constraint of Eq. 6.33e is applied and the Mode 2 constraint of Eq. 6.33f is inactivated). Initially, we consider the case where the CLBF-EMPC of Eq. 6.33 is designed using the nominal system of Eq. 6.33e for prediction, and also applied to the nominal system of Eq. 6.1. Since the prediction model and the actual process model are both the nominal system with $w(t) \equiv 0$, from the constraint of Eq. 6.33e, it is trivial to show that $W_c(\hat{x}(t_{k+1})) \leq \rho_e \leq \rho$ for the nominal system of Eq. 6.1 where again, $\hat{x}$ denotes the predicted state of the nominal system. Now we consider the case where CLBF-EMPC uses the nominal system for prediction, but is applied to the system of Eq. 6.1 subject to small bounded disturbances $|w(t)| \leq \theta$. The predicted state is still within $\mathscr{U}_{\rho_e}$ (i.e., $W_c(\hat{x}(t_{k+1})) \leq \rho_e$) based on the constraint of Eq. 6.33e.

However, by Propositions 6.3, 6.4 and Eq. 6.45, it follows that

$$\begin{aligned} W_c(x(t_{k+1})) &\leq +\alpha_4(\alpha_1^{-1}(\rho-\rho_0))|x(t_{k+1})-\hat{x}(t_{k+1})| \\ &\quad +\beta|x(t_{k+1})-\hat{x}(t_{k+1})|^2+W_c(\hat{x}(t_{k+1})) \\ &\leq \alpha_4(\alpha_1^{-1}(\rho-\rho_0))f_w(\Delta)+\beta(f_w(\Delta))^2+\rho_e \\ &\leq \rho \end{aligned} \tag{6.46}$$

Therefore, for any $x(t_k) \in \mathscr{U}_{\rho_e}$, regardless of whether the CLBF-EMPC is applied to the nominal system or the disturbed system with sufficiently small bounded disturbances, the state $x(t_{k+1})$ is always bounded in $\mathscr{U}_\rho$. Additionally, it is trivial to show that the above inequality holds for any $t \in [t_k, t_{k+1})$ if we plug in a smaller sampling period into the monotonically increasing function $f_w(\cdot)$ in Eq. 6.46.

Part 2 : In this part, we prove that if $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, the closed-loop state $x(t)$ will move towards the origin within the next sampling period (i.e., $W_c(x(t)) \leq W_c(x(t_k))$, $\forall t \in [t_k, t_{k+1})$), and will enter $\mathscr{U}_{\rho_e}$ within finite sampling steps. Since it is assumed that $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, the Mode 2 constraint of Eq. 6.33f is activated in this case and the Mode 1 constraint of Eq. 6.33e remains inactive. Similarly, we first consider the scenario that both the prediction model and the real model are the nominal system of Eq. 6.1 with $w(t) \equiv 0$. From the constraint of Eq. 6.33f and Eq. 6.13b, the following inequality is obtained:

$$\begin{aligned} \dot{W}_c(x(t_k), u(t_k)) &= \frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), u(t_k), 0) \\ &\leq \frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), \Phi(x(t_k)), 0) \\ &\leq -\alpha_3(|x(t_k)|) \end{aligned} \tag{6.47}$$

where $u(t_k)$ is the optimal input derived by the CLBF-EMPC at $t = t_k$, and applied at the next sampling period (i.e., $\forall t \in [t_k, t_{k+1})$). Under the sample-and-hold implementation of the CLBF-EMPC of Eq. 6.33, $\dot{W}_c(x(t)), \forall t \in [t_k, t_{k+1})$ is derived using results similar to Eq. 6.43 and

Eq. 6.44 by letting $w(t) = 0$:

$$\begin{aligned} \dot{W}_c(x(t), u(t_k)) &= \frac{\partial W_c(x(t))}{\partial x} F(x(t), u(t_k), 0) \\ &\leq -\alpha_3(|x(t_k)|) + L_x^{'} M\Delta + L_w^{'}|0| \end{aligned} \tag{6.48}$$

Correspondingly, we can also derive the upper bound for $\dot{W}_c(x(t), u(t_k))$ for the case that the CLBF-EMPC is designed using the nominal system of Eq. 6.1 but applied to the system of Eq. 6.1 subject to bounded disturbances. The results are shown as follows:

$$\begin{aligned} \dot{W}_c(x(t), u(t_k)) &= \frac{\partial W_c(x(t))}{\partial x} F(x(t), u(t_k), w(t)) \\ &\leq -\alpha_3(|x(t_k)|) + L_x^{'} M\Delta + L_w^{'}\theta \end{aligned} \tag{6.49}$$

Since Eq. 6.40 in Proposition 6.4 is satisfied, it implies that for both the nominal system of Eq. 6.1 and the system of Eq. 6.1 subject to bounded disturbances, $\dot{W}_c(x(t)) \leq -\varepsilon_w/\Delta$, $\forall t \in [t_k, t_{k+1})$ holds, from which we can conclude that $W_c(x(t)) \leq W_c(x(t_k))$, $\forall t \in [t_k, t_{k+1})$ and $W_c(x(t_{k+1})) \leq W_c(x(t_k)) - \varepsilon_w$ through the integral of $\dot{W}_c(x(t))$. As a result, it follows that within finite sampling steps, $W_c(x)$ will be rendered less than ρ_e, which implies that the closed-loop state $x(t)$ moves back into $\mathscr{U}_{\rho_e}$.

So far, we have proved that under the CLBF-EMPC of Eq. 6.33, whether $x(t_k) \in \mathscr{U}_{\rho_e}$ or $x(t_k) \in \mathscr{U}_{\rho} \backslash \mathscr{U}_{\rho_e}$, the state at the next sampling time $x(t_{k+1})$ is guaranteed to be bounded in $\mathscr{U}_{\rho}$. By rolling the horizon, it is trivial to show that $x(t),\ t \geq t_k \geq 0$ is always bounded in the stability and safety region $\mathscr{U}_{\rho}$, which implies that given any initial condition $x_0 \in \mathscr{U}_{\rho}$, closed-loop stability and process operational safety are guaranteed under the CLBF-EMPC of Eq. 6.33.

Part 3 : Lastly, we prove that there exists a feasible solution (e.g., the explicit stabilizing controller $\Phi(x)$ designed based on W_c with its minimum at the origin and meeting Eqs. 6.12 and 6.13 implemented in sample-and-hold) for the optimization problem of the CLBF-EMPC of Eq. 6.33 all the time. First, assuming that $x(t_k) \in \mathscr{U}_{\rho_e}$, the sample-and-hold CLBF-based control law $u(t) = \Phi(x(t_k + i\Delta)),\ i = 0, 1, \ldots, N-1$ is a feasible solution to the optimization

problem of Eq. 6.33. Specifically, it satisfies both the input constraint of Eq. 6.33d and the constraint of Eq. 6.33e because the state will move towards the origin or the saddle points $x_e \in \mathscr{U}_{\rho_e}$ under $\Phi(x)$ as shown in the CLBF-MPC of Eq. 6.27. In either case, the predicted states $\tilde{x}(t_k + i\Delta),\ i = 0, 1, \ldots, N-1$ are bounded in $\mathscr{U}_{\rho_e}$). On the other hand, if $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, the explicit stabilizing controller $u(t) = \Phi(x(t_k))$ can be directly used as a feasible solution since it meets the input constraint of Eq. 6.33d and the constraint of Eq. 6.33f.

After the optimal solution derived from the CLBF-EMPC of Eq. 6.33 is applied to the next sampling period of the system of Eq. 6.1 and the time instance is moving one sampling period forward (i.e., the rolling horizon), there again exists a feasible control action for $x(t_{k+1})$ at $t = t_{k+1}$ since $x(t_{k+1}) \in \mathscr{U}_\rho$ is guaranteed. The analysis for the two scenarios: $x(t_{k+1}) \in \mathscr{U}_{\rho_e}$ or $x(t_{k+1}) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$ follows exactly the same discussion in the last paragraph. Therefore, the optimization problem of the CLBF-EMPC of Eq. 6.33 is feasible for all $x(t) \in \mathscr{U}_\rho$ if $x_0 \in \mathscr{U}_\rho$. □

Remark 6.7. *In the formulation of the CLBF-EMPC of Eq. 6.33, ρ_e is determined by Eq. 6.45 to make $\mathscr{U}_\rho$ a forward invariant set in the presence of small bounded disturbances. Additionally, $\mathscr{U}_{\rho_e}$ is designed to include the saddle points x_e where $\partial W_c(x_e)/\partial x = 0$ such that the issue of convergence to x_e will not occur in CLBF-EMPC. Specifically, when $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, the Mode 2 constraint of Eq. 6.33f will drive the process state into $\mathscr{U}_{\rho_e}$ without having any issue of saddle points since x_e are not included in $\mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$. Furthermore, the saddle points will not be an issue either when $x(t_k) \in \mathscr{U}_{\rho_e}$ since the state attempts to move dynamically within $\mathscr{U}_{\rho_e}$ instead of converging to a saddle point in order to maximize process economic benefits under the Mode 1 constraint of Eq. 6.33e. Therefore, saddle points are handled decently under CLBF-EMPC due to the nature of EMPC that process economic performance is optimized in a consistently dynamic fashion. However, if the system is required to be operated at the origin under a tracking MPC, for example, the CLBF-MPC of Eq. 6.27, $W_c(x)$ needs to be well-designed such that x_e is a saddle point, and an additional constraint needs to be designed in MPC layer to drive the state away from x_e in case the state gets trapped in x_e.*

6.6.2 Application to a Chemical Process Example

We use the same chemical process example as in Chapter 1 to illustrate the application of CLBF-EMPC that maintains the closed-loop state within the stability and safety region in state-space. A well-mixed, non-isothermal continuous stirred tank reactor (CSTR) where an irreversible second-order exothermic reaction (from reactant A to product B) takes place is considered. The CSTR dynamic model and the description of process variables can be found in Section 1.3.1.

The states of the CSTR system are the concentration of A in the reactor (denoted by C_A) and the temperature of the reactor (denoted by T). The manipulated inputs are the inlet concentration of species A (denoted by C_{A0}) and the heat input rate (denoted by Q). The CSTR is initially operated at the steady-state $(C_{As},\ T_s) = (1.22\ kmol/m^3,\ 438\ K)$, and $(C_{A0_s}\ Q_s) = (4\ kmol/m^3,\ 0\ kJ/hr)$. Additionally, all the variables are represented in their deviation forms, i.e., the states and the inputs of the closed-loop system are $x^T = [C_A - C_{As}\ T - T_s]$ and $u^T = [\Delta C_{A0}\ \Delta Q]$, respectively, where $\Delta C_{A0} = C_{A0} - C_{A0_s}$ and $\Delta Q = Q - Q_s$. The manipulated inputs are bounded as follows: $|\Delta C_{A0}| \leq 3.5\ kmol/m^3$ and $|\Delta Q| \leq 5 \times 10^5\ kJ/hr$. The control objective is to maximize the profit of CSTR process while keeping the closed-loop state trajectories in the stability and safety region $\mathscr{U}_\rho$ using a CLBF-EMPC scheme. The objective function of the CLBF-EMPC optimizes the production rate of B: $l_e(\tilde{x}, u) = k_0 e^{-E/RT} C_A^2$.

The unsafe region $\mathscr{D}$ is defined as an open set inside the stability region (i.e., the level set of $V(x)$) where the temperature in $\mathscr{D}$ is relatively high, for this example, an ellipse described by $\mathscr{D} := \{x \in \mathbf{R}^2 \mid F(x) = (x_1 + 0.92)^2 + \frac{(x_2-42)^2}{500} < 0.06\}$. H is defined as $H := \{x \in \mathbf{R}^2 \mid F(x) < 0.07\}$, and therefore, the control barrier function $B(x)$ is designed as follows:

$$B(x) = \begin{cases} e^{\frac{F(x)}{F(x)-0.07}} - e^{-6}, \text{if } x \in H \\ -e^{-6}, \text{if } x \notin H \end{cases} \tag{6.50}$$

Then, a control Lyapunov function $V(x) = x^T Px$ is constructed with $P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix}$. Finally, the control Lyapunov-barrier function $W_c(x) = V(x) + \mu B(x) + \nu$ is constructed following the procedure in Proposition 6.1, where the parameters are determined as follows, $\rho_c = 0$, $c_1 = 0.1$, $c_2 = 1061$, $c_3 = \max_{x \in \partial H} |x|^2 = 2295$, $c_4 = \min_{x \in \partial \mathscr{D}} |x|^2 = 1370$, $\nu = \rho_c - c_1 c_4 = -160$. Hence, μ is chosen to be 1×10^9 to satisfy Eq. 6.19 and $\mathscr{U}_\rho$ with $\rho = -2.47 \times 10^6$ is the stability and safety region in the simulation. Based on the above $W_c(x)$, x_e is calculated to be a saddle point (-1.00, 47.5) in state-space. Additionally, a material constraint $\frac{1}{t_p} \int_0^{t_p} u_1(\tau) d\tau = 0$ $kmol/m^3$ is introduced to make the averaged reactant material available over a given operating period $t_p = 1.0$ hr to be 0 (in deviation from the steady-state value, C_{A0s}). The explicit Euler method with an integration time step of $h_c = 10^{-4}$ hr is applied to numerically simulate the dynamic model described by Eq. 1.1 in Section 1.3.1. The nonlinear optimization problem of the CLBF-EMPC of Eq. 6.33 is solved using the IPOPT software package [158] with the sampling period $\Delta = 10^{-2}$ hr. The closed-loop state and manipulated input profiles of the CSTR system under the CLBF-EMPC of Eq. 6.33 are shown in Fig. 6.12 and Fig. 6.13, respectively, where the dashed horizontal lines in Fig. 6.13 are the upper and lower bounds for the manipulated inputs.

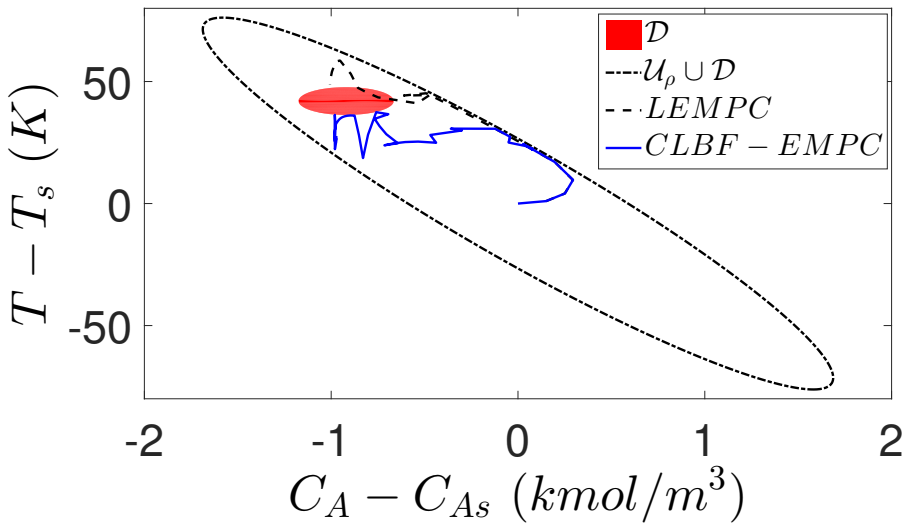


Figure 6.12: The state-space profiles for the closed-loop CSTR under LEMPC and under the CLBF-EMPC of Eq. 6.33 for an initial condition $(0,0)$.

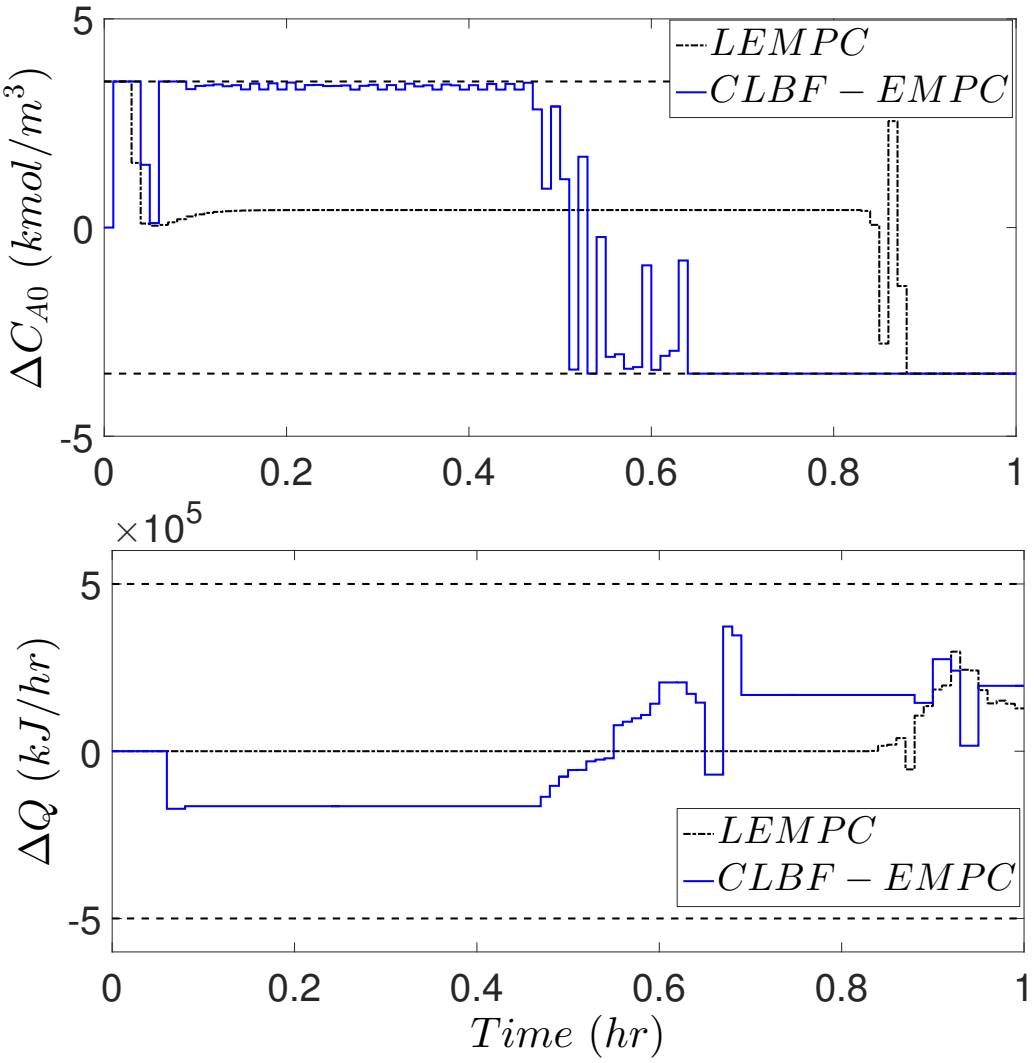


Figure 6.13: Manipulated input profiles ($u_1 = \Delta C_{A0}$, $u_2 = \Delta Q$) for the initial condition (0,0) under the CLBF-EMPC of Eq. 6.33, and under the LEMPC of Eq. 3.1.

In Fig. 6.12, it is demonstrated that the CLBF-EMPC can maintain the state of the closed-loop CSTR system within the stability and safety region (i.e., $\mathscr{U}_\rho$), while under the standard LEMPC of Eq. 3.1 that does not account for safety considerations, the closed-loop states are only guaranteed to be bounded in the stability region (i.e., $\mathscr{U}_\rho \cup \mathscr{D}$), but not within the safe region (i.e., it is possible for the trajectory to cross the red unsafe region in Fig. 6.12). In Fig. 6.13, it is demonstrated that the optimized control actions satisfy the input constraints and the material constraint. Specifically, under CLBF-EMPC, the control system consumes approximately the maximum allowable reactant ΔC_{A0} during the first 0.5 hr, and therefore has to lower the consumption at the second half hour to meet the material constraint. From $t = 0.5$ hr, the control actions also show oscillation when the closed-loop state approaches the boundary of the unsafe region $\mathscr{D}$ because the closed-loop system dynamics attempt to drive the states across the unsafe region, yet the CLBF constraint prevents this undesirable behavior.

Additionally, it is calculated that the economic benefits $L_E = \int_0^{t_p} l_e(x,u)dt$ within the entire operation period $t_p = 1$ hr under steady-state operation, and under the CLBF-EMPC are 13.9 and 16.2, respectively, from which it is shown that the CLBF-EMPC economically outperforms steady-state operation and ensures process operational safety.

6.7 Conclusions

In this chapter, CLBF-based MPC scheme was developed to optimize closed-loop performance and ensure closed-loop stability and operational safety simultaneously for nonlinear systems associated with a bounded/unbounded unsafe region. CBFs were first introduced to maintain a safe operation for nonlinear systems by avoiding undesirable regions in state-space. Subsequently, a constrained CLBF was developed for input-constrained systems by combining a CLF and a CBF together following a specific construction method. Following that, CLBF-based controllers were designed with a rigorous theoretical analysis of closed-loop stability and operational safety showing that the closed-loop state is driven to the steady-state while avoiding the unsafe region for all times. Both

the cases of bounded and unbounded unsafe regions were discussed, in which it was demonstrated a discontinuous control action was required to address the issue of convergence of the state to saddle points under continuous implementation of a stabilizing controller in the presence of a bounded unsafe region.

In order to optimize closed-loop performance while accounting for closed-loop stability and operational safety, CLBF-based MPC scheme was developed by incorporating CLBFs in the designs of stability and safety constraints. The formulations of the CLBF-MPC scheme was provided and rigorous theoretical treatments of the schemes were carried out. Based on that, a new class of economic model predictive controllers (EMPC) for nonlinear systems that account for process operational safety and economic optimality simultaneously was also developed. The effectiveness of the MPC and EMPC schemes were demonstrated using chemical process examples. Additionally, the superiority of CLBF-MPC was demonstrated through the comparison with an explicit CLBF-based controller and a standard MPC with state constraints.

Chapter 7

Machine Learning in Process Operational Safety

In Chapter 6, control Lyapunov-barrier function (CLBF)-based MPCs that ensure closed-loop stability and operational safety were developed based on a nonlinear first-principles process model. However, as such a process model is very often unavailable for complex processes in chemical industries, we have demonstrated that recurrent neural network modeling can be used to derive a data-driven process models for MPCs. Therefore, in this chapter, CLBF-MPC and CLBF-EMPC schemes that use RNN models for prediction are developed with guaranteed closed-loop stability and operational safety. Additionally, online learning of RNN models that has been discussed in Chapter 4 is also employed to update machine learning models in real-time implementation of controllers to capture the most recent process dynamics subject to time-varying disturbances. The machine learning-based control schemes are applied to a chemical reactor example to demonstrate the effectiveness of the control schemes in stabilizing systems with guaranteed safety.

7.1 Preliminaries

7.1.1 Notation

The set of real numbers is denoted by $\mathbf{R}$, and the set of nonnegative real numbers is denoted by $\mathbf{R}_+$. $\mathbf{R}^n$ is an n-dimensional real (Euclidean) space. The notation $|\cdot|$ is used to denote the Euclidean norm of a vector, and the notation $|\cdot|_Q$ denotes a weighted Euclidean norm of a vector (i.e., $|x|_Q = \sqrt{x^T Q x}$ where Q is a positive definite matrix). x^T denotes the transpose of x. The notation $L_f V(x)$ denotes the standard Lie derivative of function $V(x)$ with respect to the vector field f, i.e., $L_f V(x) := \frac{\partial V(x)}{\partial x} f$. A scalar continuous function $V : \mathbf{R}^n \to \mathbf{R}$ is proper if the set $\{x \in \mathbf{R}^n \mid V(x) \leq k\}$ is compact for all $k \in \mathbf{R}$, or equivalently, V is radially unbounded in the sense that $\lim_{|x|\to+\infty} V(x) = +\infty$ holds.

For given positive real numbers β and ε, $\mathscr{B}_\beta(\varepsilon) := \{x \in \mathbf{R}^n \mid |x-\varepsilon| < \beta\}$ is an open ball around ε with radius of β. The relative complement of the set A in B is denoted by $A \backslash B := \{x \in A, x \notin B\}$. A function $f(\cdot)$ is of class $\mathscr{C}^1$ if it is continuously differentiable. Given a set $\mathscr{D}$, the boundary, the closure, and the interior of $\mathscr{D}$ are denoted by $\partial\mathscr{D}$, $\overline{\mathscr{D}}$, and Int($\mathscr{D}$), respectively. A continuous function $\alpha : [0, a) \to \mathbf{R}_+$ is said to be of class $\mathscr{K}$ if it is strictly increasing and $\alpha(0) = 0$.

7.1.2 Class of Nonlinear Systems

The class of continuous-time nonlinear systems considered is described by the following system of first-order nonlinear ordinary differential equations:

$$\dot{x} = F(x,u,w) := f(x) + g(x)u + h(x)w,\ x(t_0) = x_0 \tag{7.1}$$

where $x \in D \subset \mathbf{R}^n$ is the state vector, $u \in U \subset \mathbf{R}^m$ is the manipulated input vector, and $w \in W$ is the disturbance vector, where $W := \{w \in \mathbf{R}^l \mid |w| \leq w_m,\ w_m \geq 0\}$. The control action constraint is defined by $u \in U := \{u_{\min} \leq u \leq u_{\max}\} \subset \mathbf{R}^m$, where u_{min} and u_{max} are the lower and upper bounds for the input vector, respectively. It is assumed that $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are sufficiently

smooth vector and matrix functions of dimensions $n \times 1$, $n \times m$, and $n \times l$, respectively, with $f(0) = 0$. Therefore, the origin is a steady-state of the nominal system of Eq. 7.1 with $w(t) \equiv 0$. The measurement of $x(t)$ is assumed to be available for feedback at each sampling time $t_k = t_0 + k\Delta$, $k = 0, 1, \ldots$, where Δ is the sampling period.

7.1.3 Stabilization Via Control Lyapunov Function

Assumption 7.1. *We assume that there exists a stabilizing feedback controller $u = \Phi(x) \in U$ for the nominal system of Eq. 7.1 with $w(t) \equiv 0$ that renders the origin of the closed-loop system under continuous implementation of the controller exponentially stable in the sense that there exists a $\mathscr{C}^1$ Lyapunov function $V : D \to \mathbf{R}_+$ such that the following inequalities hold for all x in a neighborhood D around the origin:*

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2, \tag{7.2a}$$

$$\frac{\partial V(x)}{\partial x} F(x, \Phi(x), 0) \leq -c_3|x|^2, \tag{7.2b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq c_4|x| \tag{7.2c}$$

where c_i, $i = 1, 2, 3, 4$ are positive real numbers.

The stability region Ω_ρ is designed as a level set of the Lyapunov function $V(x)$ within D, from which Eq. 7.2 is satisfied: $\Omega_\rho := \{x \in D \mid V(x) \leq \rho,\ \rho > 0\}$. It is readily shown that Ω_ρ is an invariant set since it holds that $\dot{V} \leq -c_3|x|^2$ under $u = \Phi(x) \in U$ for all $x \in \Omega_\rho$. The following control law is used to render the origin of the nominal system of Eq. 7.1 exponentially stable.

$$k_i(x) = \begin{cases} -\dfrac{p + \sqrt{p^2 + \gamma|q|^4}}{|q|^2} q_i & \text{if} \quad q \neq 0 \\ 0 & \text{if} \quad q = 0 \end{cases} \tag{7.3a}$$

$$\Phi_i(x) = \begin{cases} u_{min} & \text{if} \quad k_i(x) < u_{min} \\ k_i(x) & \text{if} \quad u_{min} \leq k_i(x) \leq u_{max} \\ u_{max} & \text{if} \quad k_i(x) > u_{max} \end{cases} \tag{7.3b}$$

where p denotes $L_f V(x)$, q_i denotes $L_{g_i} V(x)$, $q = [q_1 \cdots q_m]^T$, $f = [f_1 \cdots f_n]^T$, $g_i = [g_{i1} \cdots g_{in}]^T$, $(i = 1, 2, \cdots, m)$ and $\gamma > 0$. $k_i(x)$ of Eq. 7.3a represents the original Sontag control law without saturation. $\Phi_i(x)$ of Eq. 7.3b represents the i_{th} component of the saturated control law $\Phi(x)$ that accounts for the input constraint $u \in U$.

7.2 CLBF-MPC Using RNN models

In Chapter 6 control Lyapunov-barrier functions (CLBF) have been adopted to design model predictive controllers (MPC) for input-constrained nonlinear systems to ensure closed-loop stability and process operational safety simultaneously. In this section, a machine-learning-based CLBF-MPC is developed by taking advantage of the ensemble of RNN models that approximate the nonlinear system of Eq. 7.1. Closed-loop stability and process operational safety analysis for the system of Eq. 7.1 associated with two types of unsafe regions, i.e., bounded and unbounded sets, is also provided.

To begin with, we assume that there is a set $\mathscr{D} \subset \mathbf{R}^n$ within which it is unsafe for the system to be operated, and a safe stability region $\mathscr{U}$ that satisfies $\mathscr{U} \cap \mathscr{D} = \emptyset$ and $\{0\} \subset \mathscr{U}$, within which simultaneous closed-loop stability and process operational safety are achieved in the following sense:

Definition 7.1. *Consider the system of Eq. 7.1 and input constraints $u \in U$. If there exists a control law $u = \Phi(x) \in U$ such that for any initial state $x(t_0) = x_0 \in \mathscr{U}$, $x(t)$ remains inside $\mathscr{U}$, $\forall t \geq 0$, and the origin of the closed-loop system of Eq. 7.1 can be rendered asymptotically stable, we say that that closed-loop stability and operational safety are achieved simultaneously.*

The unsafe region is characterized based on the safety analysis of processes either from first-principles models or process operational data. Specifically, as demonstrated in Chapter 6,

there are two types of unsafe regions: 1) bounded sets, which are generally encountered in motion planning for robots and self-driving cars, and 2) unbounded sets, which are very common in chemical processes, for example, an unsafe region within which the temperature in a reactor is above a threshold that indicates an unsafe operation. In this chapter, both bounded unsafe region (denoted by $\mathscr{D}_b$) and unbounded unsafe region (denoted by $\mathscr{D}_u$) will be discussed. A CLBF-based predictive controller based on machine learning models will be developed to ensure that the closed-loop state can be driven to the steady-state and avoid the unsafe region (bounded and unbounded).

7.2.1 Stabilization and Safety via CLBF-Based Control

The definition of the CLBF of Eq. 6.12 is restated here for convenience.

Definition 7.2. *Given a set of unsafe points in state-space $\mathscr{D}$, a proper, lower-bounded and $\mathscr{C}^1$ function $W_c(x) : \mathbf{R}^n \to \mathbf{R}$ is a constrained CLBF if $W_c(x)$ has a minimum at the origin and also satisfies the following properties:*

$$W_c(x) > \rho, \ \ \forall\, x \in \mathscr{D} \subset \phi_{uc} \tag{7.4a}$$

$$L_{\hat{f}} W_c(x) < 0, \ \ \forall\, x \in \{z \in \phi_{uc} \backslash (\mathscr{D} \cup \{0\} \cup \mathbb{X}_e) \mid L_{\hat{g}} W_c(z) = 0\} \tag{7.4b}$$

$$\mathscr{U}_\rho := \{x \in \phi_{uc} \mid W_c(x) \leq \rho\} \neq \emptyset \tag{7.4c}$$

where $\rho \in \mathbf{R}$, and $\mathbb{X}_e := \{x \in \phi_{uc} \backslash (\mathscr{D} \cup \{0\}) \mid \partial W_c(x)/\partial x = 0\}$ is a set of states for the RNN model of Eq. 2.4 where $L_{\hat{f}} W_c(x) = 0$ (for $x \neq 0$) due to $\partial W_c(x)/\partial x = 0$. $\hat{f}$ and $\hat{g}$ are from the RNN model in the form of Eq. 2.5. Additionally, in this section, we consider the case that the safe operating region is a level set of W_c, (i.e., $\mathscr{U}_\rho$) only, and therefore, the original definition of CLBF of Eq. 6.12 can be simplified to Eq. 7.4 according to the discussion in Remark 6.1. A feedback control law $u = \Phi_{nn}(x) \in U$ that renders the origin exponentially stable within an open neighborhood ϕ_{uc} that includes the origin in its interior is assumed to exist for the RNN system of Eq. 2.4 (also in the form of Eq. 2.5) in the sense that there exists a $\mathscr{C}^1$ constrained control Lyapunov-barrier function

$W_c(x)$ that has a minimum at the origin and satisfies the following inequalities $\forall x \in \phi_{uc}$:

$$\hat{c}_1|x|^2 \leq W_c(x) - \rho_0 \leq \hat{c}_2|x|^2, \tag{7.5a}$$

$$\begin{aligned} &\frac{\partial W_c(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x)) \leq -\hat{c}_3|x|^2, \forall x \in \phi_{uc} \backslash \mathscr{B}_\delta(x_e) \\ &\frac{\partial W_c(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x)) \leq 0, \forall x \in \mathscr{B}_\delta(x_e) \end{aligned} \tag{7.5b}$$

$$\left| \frac{\partial W_c(x)}{\partial x} \right| \leq \hat{c}_4|x| \tag{7.5c}$$

where $\hat{c}_j(\cdot)$, $j = 1,\ 2,\ 3,\ 4$ are positive real numbers, $W_c(0) = \rho_0$ is the global minimum value of $W_c(x)$ in ϕ_{uc}, and $\mathscr{B}_\delta(x_e)$ is a small neighborhood around $x_e \in \mathbb{X}_e$. $F_{nn}(x,u)$ is the RNN system of Eq. 2.4. It is noted that $\frac{\partial W_c(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x)) \leq -\hat{c}_3|x|^2$ does not hold for $x \in \mathscr{B}_\delta(x_e)$ since $\frac{\partial W_c(x)}{\partial x}$ is close to zero in a neighborhood around the stationary point x_e, where $\frac{\partial W_c(x)}{\partial x} = 0$. The set ϕ_{uc} is characterized using numerical simulations as a set of states in the state-space where Eq. 7.5 is satisfied. Additionally, by continuity and the smoothness assumed for f, g and h in the nonlinear system of Eq. 7.1, there exist positive constants $M, L_x, L_w, L_x^{'}, L_w^{'}$ such that the following inequalities hold for all $x, x' \in \mathscr{U}_\rho, u \in U$, and $w \in W$:

$$|F(x,u,w)| \leq M \tag{7.6a}$$

$$|F(x,u,w) - F(x',u,0)| \leq L_x|x - x'| + L_w|w| \tag{7.6b}$$

$$\left| \frac{\partial W_c(x)}{\partial x} F(x,u,w) - \frac{\partial W_c(x')}{\partial x} F(x',u,0) \right| \leq L_x^{'}|x - x'| + | \leq L_w^{'}|w_m| \tag{7.6c}$$

The universal Sontag controller of Eq. 7.3 with $W_c(x)$ replacing the Lyapunov function $V(x)$ can be used as an example of the stabilizing control law $\Phi_{nn}(x)$ associated with CLBFs. It should be noted that the CLBF of Eq. 7.4 and the set ϕ_{uc} are designed based on the RNN model of Eq. 2.4 (also in the form of Eq. 2.5, i.e., $\dot{x} = \hat{f}(x) + \hat{g}(x)u$) since the nonlinear system of Eq. 7.1 is assumed to be unknown. A constrained CLBF that satisfies all the conditions in Eq. 7.4 can be developed by first designing a CLF and a CBF separately, and then combining them together via the construction

method in Section 6.4.2.

Consider the RNN model of Eq. 2.4 (also in the form of Eq. 2.5) with a constrained CLBF $W_c(x)$. Simultaneous closed-loop stability and safety can be derived for both a bounded unsafe region $\mathscr{D}_b$ and an unbounded unsafe region $\mathscr{D}_u$ following the similar analysis that has been performed for the nominal system of Eq. 7.1 in Section 6.4.1.3 (see Theorem 6.2 and Theorem 6.3). Specifically, it is noted that in the case of a bounded unsafe set, there exist stationary points (other than the origin) in state-space (i.e., $\mathbb{X}_e$ in Eq. 7.4b), and thus, a continuous controller cannot render the origin exponentially stable. This issue can be addressed by designing the stationary points to be saddle points and then implementing discontinuous control actions at saddle points to drive the state away from them in the direction of decreasing $W_c(x)$. However, in the presence of an unbounded unsafe region, the origin is the unique stationary point in state-space, thereby closed-loop stability and process operational safety can be readily derived under the controller $u = \Phi_{nn}(x) \in U$. The following theorem provides sufficient conditions under which closed-loop stability and process operational safety are achieved simultaneously for the RNN system of Eq. 2.4 under the control law designed based on a constrained CLBF of Eq. 7.4.

Theorem 7.1. *Consider that a constrained CLBF $W_c(x)$: $\mathbf{R}^n \rightarrow \mathbf{R}$ that has a minimum at the origin and meets the conditions of Eq. 7.4, exists for the RNN system of Eq. 2.4. The controller $u = \Phi_{nn}(x) \in U$ that satisfies Eq. 7.5 guarantees that the closed-loop state stays in $\mathscr{U}_\rho$ for all times for any $x_0 \in \mathscr{U}_\rho$. Additionally, the origin can be rendered exponentially stable under $u = \Phi_{nn}(x) \in U$, for all $x_0 \in \mathscr{U}_\rho$ in the presence of an unbounded unsafe region $\mathscr{D}_u$; however, discontinuous control actions $u = \bar{u}(x) \in U$ that decrease $W_c(x)$ are required at saddle points x_e to ensure exponential stability of the origin in the presence of a bounded unsafe region $\mathscr{D}_b$ in state-space.*

Proof. To demonstrate that the state is bounded in the safe operating region $\mathscr{U}_\rho$ for all times, we need to show that there exists a controller $u = \Phi_{nn}(x) \in U$ such that $\dot{W}_c \leq 0$ holds for all $x \in \mathscr{U}_\rho$. This has been proven in Theorem 6.2 by showing that the universal Sontag controller of Eq. 7.3 with $W_c(x)$ replacing the Lyapunov function $V(x)$ can be utilized as $\Phi_{nn}(x)$. Additionally, since $\mathscr{U}_\rho$ is characterized as a level set of $W_c(x)$ in ϕ_{uc} within which Eq. 7.5 is satisfied, we can

further demonstrate that the origin can be rendered exponentially stable under $u = \Phi_{nn}(x) \in U$. The issue of saddle points in the presence of a bounded unsafe region is handled by discontinuous control actions $\bar{u}(x)$ (i.e., $\bar{u}(x) \neq \Phi_{nn}(x)$). The detailed proofs for both bounded and unbounded unsafe regions follow closely to those for Theorem 6.2 and Theorem 6.3 in Section 6.4.1.3, and are omitted here. □

Remark 7.1. *As we assume that the nonlinear system of Eq. 7.1 is unknown, the CLBF of Eq. 7.4 and the safe operating region $\mathscr{U}_\rho$ are characterized based on the RNN system of Eq. 2.4. Theorem 7.1 is established to demonstrate that closed-loop stability and operational safety are achieved for the RNN system of Eq. 2.4 via a stabilizing controller $u = \Phi_{nn}(x) \in U$ that is defined with respect to the CLBF of Eq. 7.4. In the following section, we will demonstrate that the CLBF-based controller $u = \Phi_{nn}(x) \in U$ also guarantees simultaneous closed-loop stability and operational safety for the nonlinear system of Eq. 7.1 provided that the modeling error between the nonlinear system of Eq. 7.1 and the RNN system of Eq. 2.4 is sufficiently small.*

7.2.2 CLBF-based MPC Using an Ensemble of RNN Models

This section presents the formulation of the CLBF-based MPC (CLBF-MPC) that incorporates an ensemble of RNN models for predicting future states. We first demonstrate that the stability and safety properties in Theorem 7.1 hold for the nominal system of Eq. 7.1 (i.e., $w(t) \equiv 0$) under the CLBF-based controller $u = \Phi_{nn}(x) \in U$ that is designed to stabilize the RNN system of Eq. 2.4 with guaranteed safety. Subsequently, the CLBF-MPC is developed to drive the state to a small neighborhood around the origin while optimizing process performance under sample-and-hold implementation of control actions. To proceed, the following proposition is first developed to obtain an upper bound for the error between the states predicted by the RNN model of Eq. 2.4 and the states of the nonlinear process of Eq. 7.1 in the presence of bounded disturbances (i.e., $|w(t)| \leq w_m$) and a bounded modeling error (i.e., $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma|x| \leq \nu_m$).

Proposition 7.1. *Consider the nonlinear system $\dot{x} = F(x,u,w)$ of Eq. 7.1 in the presence of bounded disturbances $|w(t)| \leq w_m$. Assuming that the RNN model $\dot{\hat{x}} = F_{nn}(\hat{x},u)$ of Eq. 2.4 has*

the same initial condition $x_0 = \hat{x}_0 \in \mathscr{U}_\rho$ as the nonlinear system of Eq. 7.1, there exists a class $\mathscr{K}$ function $f_w(\cdot)$ and a positive constant κ such that the following inequalities hold $\forall x, \hat{x} \in \mathscr{U}_\rho$ and $w(t) \in W$:

$$|x(t) - \hat{x}(t)| \leq f_w(t) := \frac{L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1) \tag{7.7a}$$

$$W_c(x) \leq W_c(\hat{x}) + \frac{\hat{c}_4\sqrt{\rho - \rho_0}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{7.7b}$$

Proof. Let $e(t) = x(t) - \hat{x}(t)$ denote the error vector between the solutions of the system $\dot{x} = F(x,u,w)$ and the RNN model $\dot{\hat{x}} = F_{nn}(\hat{x},u)$. The time-derivative of $e(t)$ is obtained as follows:

$$\begin{aligned} |\dot{e}| &= |F(x,u,w) - F_{nn}(\hat{x},u)| \\ &\leq |F(x,u,w) - F(\hat{x},u,0)| + |F(\hat{x},u,0) - F_{nn}(\hat{x},u)| \end{aligned} \tag{7.8}$$

Using Eq. 7.6b, the upper bound for the first term of Eq. 7.8 is derived by the following inequality for all $x, \hat{x} \in \mathscr{U}_\rho$ and $w(t) \in W$:

$$\begin{aligned} |F(x,u,w) - F(\hat{x},u,0)| &\leq L_x|x(t) - \hat{x}(t)| + L_w|w(t)| \\ &\leq L_x|x(t) - \hat{x}(t)| + L_w w_m \end{aligned} \tag{7.9}$$

Additionally, it is noticed that the second term of Eq. 7.8 represents the modeling error (i.e., $|\nu| = |F(\hat{x},u,0) - F_{nn}(\hat{x},u)|$), and is bounded by $|\nu| \leq \nu_m$. Therefore, the upper bound for $\dot{e}(t)$ in Eq. 7.8 is obtained as follows:

$$\begin{aligned} |\dot{e}(t)| &\leq L_x|x(t) - \hat{x}(t)| + L_w|w_m| + \nu_m \\ &\leq L_x|e(t)| + L_w|w_m| + \nu_m \end{aligned} \tag{7.10}$$

Given the zero initial condition (i.e., $e(0) = 0$), the upper bound for $|e(t)|$ is derived for all $x(t), \hat{x}(t) \in \mathscr{U}_\rho$ and $|w(t)| \leq w_m$ as follows:

$$|e(t)| = |x(t) - \hat{x}(t)| \leq f_w(t) \tag{7.11}$$

where

$$f_w(t) := \frac{L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1)$$

Moreover, since $W_c(x)$ is continuous and bounded on compact sets, the following inequality is derived based on the Taylor series expansion of $W_c(x)$ around $\hat{x}$, $\forall x, \hat{x} \in \mathscr{U}_\rho$:

$$W_c(x) \leq W_c(\hat{x}) + \frac{\partial W_c(\hat{x})}{\partial x}|x - \hat{x}| + \kappa |x - \hat{x}|^2 \tag{7.12}$$

where κ is a positive real number and the term $\kappa|x-\hat{x}|^2$ is used to bound the high order terms of the Taylor series of $W_c(x)$, $\forall x, \hat{x} \in \mathscr{U}_\rho$. The following inequality is derived using Eq. 7.5a, Eq. 7.5c and Eq. 7.11:

$$\begin{aligned} W_c(x) &\leq W_c(\hat{x}) + \frac{\hat{c}_4\sqrt{\rho - \rho_0}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa |x - \hat{x}|^2 \\ &\leq W_c(\hat{x}) + \frac{\hat{c}_4\sqrt{\rho - \rho_0}}{\sqrt{\hat{c}_1}} f_w(t) + \kappa f_w(t)^2 \end{aligned} \tag{7.13}$$

This completes the proof of Proposition 7.1. □

7.2.2.1 CLBF-based control using RNN models

The following propositions are developed to demonstrate that the controller $u = \Phi_{nn}(x) \in U$ designed for the RNN model of Eq. 2.4 is able to maintain the state of the nominal system of Eq. 7.1 within the safe operating region $\mathscr{U}_\rho$ provided that the modeling error is sufficiently small. We first consider the case of an unbounded unsafe region, for which exponential stability is achieved for the closed-loop nominal system of Eq. 7.1 under $u = \Phi_{nn}(x) \in U$.

Proposition 7.2. *Consider the nominal system of Eq. 7.1 (i.e., $w(t) \equiv 0$) with an unbounded unsafe region $\mathscr{D}_u$ under the feedback controller $u = \Phi_{nn}(x) \in U$ that satisfies Eq. 7.5 for all $x \in \mathscr{U}_\rho$. If there exists a positive real number $\gamma < \hat{c}_3/\hat{c}_4$ such that for all $x \in \mathscr{U}_\rho$ and $u \in U$, the modeling error between the RNN model of Eq. 2.4 and the nonlinear system of Eq. 7.1 is constrained by $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma|x|$, then the stability and safety properties in Theorem 7.1 also hold for the nominal closed-loop system of Eq. 7.1 under $u = \Phi_{nn}(x) \in U$.*

Proof. To demonstrate that the origin of the nominal system of Eq. 7.1 (i.e., $w(t) \equiv 0$) can be rendered exponentially stable under $u = \Phi_{nn}(x) \in U$, we prove that there exists a positive real number $\tilde{c}_3$ such that $\frac{\partial W_c(x)}{\partial x} F(x, \Phi_{nn}(x), 0) \leq -\tilde{c}_3 |x|^2$, $\forall x \in \mathscr{U}_\rho$ holds. It is noted that in the presence of an unbounded unsafe region, there is no saddle point within the safe operation region $\mathscr{U}_\rho$. Therefore, the time-derivative of W_c is derived as follows using Eq. 7.5b and Eq. 7.5c:

$$\begin{aligned}
\dot{W}_c =& \frac{\partial W_c(x)}{\partial x} F(x, \Phi_{nn}(x), 0) \\
=& \frac{\partial W_c(x)}{\partial x} (F_{nn}(x, \Phi_{nn}(x)) + F(x, \Phi_{nn}(x), 0) - F_{nn}(x, \Phi_{nn}(x))) \\
\leq& -\hat{c}_3 |x|^2 + \hat{c}_4 |x| (F(x, \Phi_{nn}(x), 0) - F_{nn}(x, \Phi_{nn}(x))) \\
\leq& -\hat{c}_3 |x|^2 + \hat{c}_4 \gamma |x|^2
\end{aligned} \tag{7.14}$$

Let $\tilde{c}_3 = -\hat{c}_3 + \hat{c}_4 \gamma$. It is obtained that $\dot{W}_c \leq -\tilde{c}_3 |x|^2 \leq 0$ if γ is chosen to satisfy $\gamma < \hat{c}_3 / \hat{c}_4$. Therefore, following the proof of closed-loop stability and safety for the RNN system of Eq. 2.4 in Theorem 7.1, the controller $u = \Phi_{nn}(x) \in U$ can drive the state of the nominal system of Eq. 7.1 to the origin while avoiding the unbounded unsafe region $\mathscr{D}_u$ for all times. This completes the proof of simultaneous closed-loop stability and operational safety for any initial condition x_0 in the safe operating region $\mathscr{U}_\rho$. □

The following proposition is developed to provide sufficient conditions under which simultaneous closed-loop stability and process operational safety are guaranteed for the nominal system of Eq. 7.1 with a bounded unsafe region $\mathscr{U}_b$ accounting for the existence of saddle points x_e in the safe operating region $\mathscr{U}_\rho$.

Proposition 7.3. *Consider the nominal system of Eq. 7.1 with a bounded unsafe region $\mathscr{D}_b$ under the controller $u = \Phi_{nn}(x) \in U$ that satisfies Eq. 7.5 for all $x \in \mathscr{U}_\rho$. If there exists a positive real number $\gamma < \hat{c}_3 / \hat{c}_4$ such that for all $x \in \mathscr{U}_\rho$ and $u \in U$, the modeling error is constrained by $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma |x|$, and Eq. 7.15 is satisfied under discontinuous control actions $u =$*

$\bar{u}(x) \in U$ *when* $x(t_k) = \hat{x}(t_k) \in \mathscr{B}_\delta(x_e)$,

$$W_c(\hat{x}(t)) < W_c(\hat{x}(t_k)) - f_e(t - t_k),\ \forall t > t_k \tag{7.15}$$

where

$$f_e(t - t_k) := \frac{\hat{c}_4\sqrt{\rho - \rho_0}}{\sqrt{\hat{c}_1}} f_w(t - t_k) - \kappa f_w(t - t_k)^2$$

and $f_w(t)$ *is given in Eq. 7.11, then the stability and safety properties in Theorem 7.1 also hold for the nominal closed-loop system of Eq. 7.1 with a bounded unsafe region* $\mathscr{D}_b$ *under* $u = \Phi_{nn}(x) \in U$ *and* $u = \bar{u}(x) \in U$.

Proof. Since there exist saddle points x_e in the safe operating region $\mathscr{U}_\rho$ in the presence of a bounded unsafe region, the origin of the nominal system of Eq. 7.1 (i.e., $w(t) \equiv 0$) cannot be rendered exponentially stable under the continuous controller $u = \Phi_{nn}(x) \in U$. To address the issue of saddle points x_e, another set of control actions $\bar{u}$ will be applied within a neighborhood around x_e to drive the state away from saddle points and towards the origin. Specifically, in the presence of a bounded unsafe region, it is readily shown that Eq. 7.14 still holds for all $x \in \mathscr{U}_\rho \backslash \mathscr{B}_\delta(x_e)$ since $\frac{\partial W_c(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x)) \leq -\hat{c}_3|x|^2$ is satisfied in $\mathscr{U}_\rho \backslash \mathscr{B}_\delta(x_e)$. This implies that in the presence of a bounded unsafe region, the controller $u = \Phi_{nn}(x) \in U$ that is designed to achieve closed-loop stability and safety for the RNN model of Eq. 2.4 is also able to maintain the state of the closed-loop system of Eq. 7.1 within $\mathscr{U}_\rho$ for all times.

Subsequently, we prove that the discontinuous control actions $u = \bar{u}(x) \in U$ that are designed for the RNN model of Eq. 2.4 around saddle points can drive the state of the nonlinear system of Eq. 7.1 away from saddle points in the direction of decreasing $W_c(x)$. Proposition 7.1 has established that starting from the same initial condition, the error between the states of the RNN system of Eq. 2.4 and of the nonlinear system of Eq. 7.1 is bounded under the same control actions, and therefore, the evolution of $W_c(x)$ based on the state of the nominal system of Eq. 7.1 is also bounded by Eq. 7.13 accounting for the modeling error and bounded disturbances. Assuming that the state enters a neighborhood around the saddle points at $t = t_k$ (i.e., $\hat{x}(t_k) = x(t_k) \in \mathscr{B}_\delta(x_e)$), if

the discontinuous control actions $\bar{u}(\hat{x})$ that are determined for the RNN model of Eq. 2.4 satisfy Eq. 7.15 for all $x \in \mathscr{B}_\delta(x_e)$, the following inequality can be derived from Eq. 7.13 and Eq. 7.15 to show that the value of $W_c(x)$ based on the state of the nonlinear system of Eq. 7.1 is guaranteed to decrease $\forall t > t_k$:

$$\begin{aligned} W_c(x(t)) &\leq W_c(\hat{x}(t)) + \frac{\hat{c}_4\sqrt{\rho-\rho_0}}{\sqrt{\hat{c}_1}} f_w(t-t_k) + \kappa f_w(t-t_k)^2, \\ &< W_c(\hat{x}(t_k)) \end{aligned} \tag{7.16}$$

Therefore, the state of the nonlinear system of Eq. 7.1 can escape from saddle points under the discontinuous control actions $u = \bar{u}(x) \in U$ that are designed for the RNN system of Eq. 2.4 provided that the decreasing rate of $W_c(x)$ of Eq. 7.15 is satisfied. This implies that for any initial condition $x_0 \in \mathscr{U}_\rho$, the closed-loop state of the nonlinear system of Eq. 7.1 can be driven to the origin while avoiding the bounded unsafe region $\mathscr{D}_b$ under the controllers $u = \Phi_{nn}(x) \in U$ and $u = \bar{u}(x) \in U$. □

Remark 7.2. *From Proposition 7.2 and Proposition 7.3, it is demonstrated that the controller $u = \Phi_{nn}(x) \in U$ that is designed to stabilize the RNN system of Eq. 2.4 (i.e., $\dot{\hat{x}} = F_{nn}(\hat{x}, u)$) guarantees simultaneous closed-loop stability and operational safety for the nominal system of Eq. 7.1 (i.e., $w(t) \equiv 0$). Specifically, in the case of an unbounded unsafe region, the state of the nominal system of Eq. 7.1 is bounded in the safe operating region $\mathscr{U}_\rho$ for all times and the origin can be rendered exponentially stable under $u = \Phi_{nn}(x) \in U$. However, to ensure closed-loop stability and operational safety for the nominal system of Eq. 7.1 in the presence of a bounded unsafe region, in addition to the controller $u = \Phi_{nn}(x) \in U$ that is applied everywhere except the neighborhood around saddle points (i.e., $\mathscr{B}_\delta(x_e)$), a set of discontinuous control actions $u = \bar{u}(x) \in U$ that satisfy Eq. 7.15 is required for the state in $\mathscr{B}_\delta(x_e)$.*

7.2.2.2 Sample-and-hold implementation of CLBF-based controller

In this section, we present the stability properties of the CLBF-based controllers $u = \Phi_{nn}(x) \in U$ and $u = \bar{u}(x) \in U$ (for a bounded unsafe region) for the nonlinear system of Eq. 7.1 accounting for bounded disturbances (i.e., $|w(t)| \leq w_m$) and sample-and-hold implementation of the control actions. To proceed, we need the following proposition to demonstrate that under the controllers $u = \Phi_{nn}(x) \in U$ and $u = \bar{u}(x) \in U$ implemented in a sample-and-hold fashion, i.e., $u(t) = u(t_k)$, $\forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$ and Δ is the sampling period, the closed-loop state $x(t)$ of the nonlinear system of Eq. 7.1 is bounded in $\mathscr{U}_\rho$ for all times, and will be ultimately driven to a small neighborhood $\mathscr{U}_{\rho_{min}}$ around the origin.

Proposition 7.4. *Consider the system of Eq. 7.1 under the sample-and-hold implementation of the controller $u = \Phi_{nn}(x) \in U$ that meets the conditions of Eq. 7.5. If Eq. 7.15 is satisfied under the controller $u = \bar{u}(x) \in U$ in a sample-and-hold fashion for $x \in \mathscr{B}_\delta(x_e)$, and there exist $\varepsilon_w > 0$, $\Delta > 0$ and $\rho > \rho_{min} > \rho_{nn} > \rho_s$ that satisfy*

$$-\frac{\tilde{c}_3}{\hat{c}_2}(\rho_s - \rho_0) + L_x^{'} M \Delta + L_w^{'} w_m \leq -\varepsilon_w \tag{7.17}$$

and

$$\rho_{nn} := \max\{W_c(\hat{x}(t+\Delta)) \mid \hat{x}(t) \in \mathscr{U}_{\rho_s}, u \in U\} \tag{7.18a}$$

$$\rho_{min} \geq \rho_{nn} + f_e(\Delta) \tag{7.18b}$$

where $f_e(t)$ is given by Eq. 7.15, then for any $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_s}$, $W_c(x(t))$ based on the state of the nonlinear system of Eq. 7.1 is guaranteed to decrease within every sampling period, and thus, can be bounded in $\mathscr{U}_\rho$ for all times and ultimately bounded in $\mathscr{U}_{\rho_{min}}$.

Proof. Assuming $x(t_k) = \hat{x}(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_s}$, the time-derivative of $W_c(x)$ for the nonlinear system of

Eq. 7.1 in the presence of bounded disturbances (i.e., $|w| \leq w_m$) is derived as follows:

$$\begin{aligned}\dot{W}_c(x(t)) =& \frac{\partial W_c(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\ =& \frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) + \frac{\partial W_c(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\ & - \frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0)\end{aligned} \tag{7.19}$$

Based on Eq. 7.5b, Eq. 7.14 and the Lipschitz condition in Eq. 7.6, the following inequality is obtained for $\dot{W}_c(x(t))$ for all $t \in [t_k, t_{k+1})$ and $x(t_k) \in \mathscr{U}_\rho \backslash (\mathscr{U}_{\rho_s} \cup \mathscr{B}_\delta(x_e))$:

$$\begin{aligned}\dot{W}_c(x(t)) \leq & -\frac{\tilde{c}_3}{\hat{c}_2}(\rho_s - \rho_0) + \frac{\partial W_c(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\ & - \frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \\ \leq & -\frac{\tilde{c}_3}{\hat{c}_2}(\rho_s - \rho_0) + L_x^{'} |x(t) - x(t_k)| + L_w^{'} |w| \\ \leq & -\frac{\tilde{c}_3}{\hat{c}_2}(\rho_s - \rho_0) + L_x^{'} M \Delta + L_w^{'} w_m\end{aligned} \tag{7.20}$$

It is noted that Eq. 7.20 does not hold for $x \in \mathscr{B}_\delta(x_e)$ since Eq. 7.14 may not hold in the neighborhood around saddle points where $\frac{\partial W_c(x)}{\partial x}$ is close to zero. If Eq. 7.17 is satisfied, we can obtain the following inequality based on Eq. 7.20 for all $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_s}$ and $t \in [t_k, t_{k+1})$:

$$\dot{W}_c(x(t)) \leq -\varepsilon_w \tag{7.21}$$

From Eq. 7.21, the boundedness of the state of the closed-loop system of Eq. 7.1 in the safe operating region $\mathscr{U}_\rho$ is obtained under the sample-and-hold implementation of $u = \Phi_{nn}(x) \in U$ for any initial condition $x_0 \in \mathscr{U}_\rho$.

Additionally, to ensure that the state of the nonlinear system of Eq. 7.1 moves towards the origin and ultimately enters a small neighborhood $\mathscr{U}_{\rho_s}$ around the origin instead of converging to saddle points, the controller $u = \bar{u}(x(t_{k+i})) \in U$, $\forall t \in [t_{k+i},\ t_{k+i+1})$, $i = 0, 1, 2, ...$ is applied to drive the state away from saddle points when $x(t_k) = \hat{x}(t_k) \in \mathscr{B}_\delta(x_e)$. If Eq. 7.15 is satisfied

under the sample-and-hold implementation of $u = \bar{u}(\hat{x}) \in U$, it is demonstrated from Eq. 7.16 in Proposition 7.3 that $W_c(x(t)) < W_c(x(t_k))$ holds for the nonlinear system of Eq. 7.1, $\forall t > t_k$. Therefore, $W_c(x)$ will keep decreasing until the state of the nonlinear system of Eq. 7.1 leaves the neighborhood around saddle points. After that, the controller $u = \Phi_{nn}(x) \in U$ will be applied again to drive the state towards the origin.

It remains to show that once the state enters $\mathscr{U}_{\rho_s}$ (i.e., $x(t_k) = \hat{x}(t_k) \in \mathscr{U}_{\rho_s}$), it is bounded in $\mathscr{U}_{\rho_{min}}$ for the remaining time $t \geq t_k$. According to the definition of $\mathscr{U}_{\rho_{nn}}$ in Eq. 7.18a, it is shown that $\mathscr{U}_{\rho_{nn}}$ is the largest level set of $W_c(\hat{x})$ that the state of the RNN system of Eq. 2.4 can reach within one sampling period if starting from $\mathscr{U}_{\rho_s}$. Additionally, $\mathscr{U}_{\rho_{min}}$ of Eq. 7.18b is the corresponding largest level set of $W_c(x)$ based on the state of the nonlinear system of Eq. 7.1 when the RNN state $\hat{x}$ is bounded in $\mathscr{U}_{\rho_{nn}}$. Since $\dot{W}_c \leq -\varepsilon_w$ may not hold for the state in $\mathscr{U}_{\rho_s}$ under the sample-and-hold implementation of $u = \Phi_{nn}(x) \in U$, the sets $\mathscr{U}_{\rho_{nn}}$ and $\mathscr{U}_{\rho_{min}}$ are characterized to guarantee that the states of the RNN system of Eq. 2.4 and of the nonlinear system of Eq. 7.1 are bounded in the neighborhoods around the origin that are slightly larger than $\mathscr{U}_{\rho_s}$. Additionally, $\mathscr{U}_{\rho_{nn}}$ can be characterized from extensive open-loop simulations for all $u \in U$ and $x \in \mathscr{U}_{\rho_s}$. Subsequently, $\mathscr{U}_{\rho_{min}}$ of Eq. 7.18b is characterized based on $\mathscr{U}_{\rho_{nn}}$ to account for the impact of modeling error and bounded disturbances within one sampling period. This completes the proof of Proposition 7.4 by showing that the state of the nonlinear system of Eq. 7.1 with bounded disturbances (i.e., $|w(t)| \leq w_m$) can be maintained in the safe operating region $\mathscr{U}_{\rho}$ for all times, and ultimately be bounded in $\mathscr{U}_{\rho_{min}}$ under the sample-and-hold implementation of $u = \Phi_{nn}(x) \in U$ and $u = \bar{u}(x) \in U$. □

7.2.2.3 Formulation of CLBF-MPC

The CLBF-MPC design is represented by the following optimization problem [174]:

$$\min_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} l_t(\tilde{x}(t),u(t))dt \tag{7.22a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = \frac{1}{N_e}\sum_{j=1}^{N_e} F_{nn}^{j}(\tilde{x}(t),u(t)) \tag{7.22b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{7.22c}$$

$$u(t) \in U,\ \forall\, t \in [t_k, t_{k+N}) \tag{7.22d}$$

$$\dot{W}_c(x(t_k),u(t_k)) \leq \dot{W}_c(x(t_k),\Phi_{nn}(t_k)),\ \text{ if } W_c(x(t_k)) > \rho_{nn} \text{ and } x(t_k) \notin \mathscr{B}_\delta(x_e) \tag{7.22e}$$

$$W_c(\tilde{x}(t)) \leq \rho_{nn},\ \forall\, t \in [t_k, t_{k+N}),\ \text{ if } W_c(x(t_k)) \leq \rho_{nn} \tag{7.22f}$$

$$W_c(\tilde{x}(t)) < W_c(x(t_k)) - f_e(t-t_k),\ \forall\, t \in (t_k, t_{k+N}),\ \text{ if } x(t_k) \in \mathscr{B}_\delta(x_e) \tag{7.22g}$$

where $\tilde{x}(t)$ is the predicted state trajectory, $S(\Delta)$ is the set of piecewise constant functions with period Δ, and N is the number of sampling periods in the prediction horizon. $\dot{W}_c(x,u)$ is used to represent $\frac{\partial W_c(x)}{\partial x}F_{nn}(x,u)$. The cost function $l_t(\tilde{x}(t),u(t))$ is generally in a quadratic form that has the minimum value at the equilibrium of the system of Eq. 7.1 : $|\tilde{x}(t)|^2_{Q_L} + |u(t)|^2_{R_L}$, where Q_L and R_L are positive definite matrices. The predicted states $\tilde{x}(t)$, $t \in [t_k, t_{k+N})$ are calculated by taking the average of an ensemble of RNN models F_{nn}^{j}, $j = 1,...,N_e$ in Eq. 7.22b, where N_e is the number of RNN models in the ensemble. The objective function of Eq. 7.22a is the time integral of $l_t(\tilde{x}(t),u(t))$ over the prediction horizon. The input constraints of Eq. 7.22d are applied over the entire prediction horizon. The state measurement of Eq. 7.22c at $t = t_k$ is taken as the initial condition for the RNN models of Eq. 7.22b. The constraints of Eqs. 7.22e-7.22g in the CLBF-MPC optimization problem are utilized to ensure closed-loop stability and process operational safety. Specifically, the constraint of Eq. 7.22e forces $W_c(\tilde{x})$ along the predicted state trajectories to decrease at least at the rate under the CLBF-based controller $u = \Phi_{nn}(x) \in U$ when $W_c(x(t_k)) > \rho_{nn}$ and $x(t_k) \notin \mathscr{B}_\delta(x_e)$. If $W_c(x(t_k)) \leq \rho_{nn}$, the constraint of Eq. 7.22f is activated to maintain the predicted state of the RNN system within $\mathscr{U}_{\rho_{nn}}$ such that the closed-loop state of the

nonlinear system of Eq. 7.1 is bounded $\mathscr{U}_{\rho_{min}}$. Additionally, if $x(t_k) \in \mathscr{B}_\delta(x_e)$, the constraint of Eq. 7.22g decreases $W_c(x)$ over the prediction horizon such that the state can escape from saddle points x_e within finite sampling steps. The state measurements of the closed-loop system of Eq. 7.1 are assumed to be available at each sampling time. After the CLBF-MPC optimization problem of Eq. 7.22 solves the optimal solution $u^*(t)$, only the first control action of $u^*(t)$ is sent to the control actuators to be applied over the next sampling period. Then, the horizon will be rolled forward one sampling time, and at the next instance of time $t_{k+1} := t_k + \Delta$, the optimization problem is solved again. Additionally, since the CLBF-MPC optimization problem of Eq. 7.22 is based on the state of the RNN model of Eq. 2.4 only, we use x instead of $\hat{x}$ to represent the RNN state in CLBF-MPC to simplify the notations.

The theorem below is established to demonstrate that under the CLBF-MPC of Eq. 7.22, closed-loop stability and process operational safety are achieved simultaneously for the nonlinear system of Eq. 7.1 in the sense that the closed-loop state is bounded in the safe operating region $\mathscr{U}_\rho$ for all times, and is ultimately bounded in $\mathscr{U}_{\rho_{min}}$.

Theorem 7.2. *Consider the system of Eq. 7.1 with a constrained CLBF W_c that satisfies Eq. 7.4 and has a minimum at the origin. Given any initial state $x_0 \in \mathscr{U}_\rho$, it is guaranteed that the CLBF-MPC optimization problem of Eq. 7.22 can be solved with recursive feasibility for all times. Additionally, under the sample-and-hold implementation of CLBF-MPC based on an ensemble of RNN models that satisfy $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma|x| \leq \nu_m$ and the conditions in Proposition 7.4, it is guaranteed that for any $x_0 \in \mathscr{U}_\rho$, the state is bounded in $\mathscr{U}_\rho$, $\forall\, t \geq 0$, and ultimately converges to $\mathscr{U}_{\rho_{min}}$ as $t \to \infty$.*

Proof. The proof consists of two parts. The first part presents the proof of recursive feasibility of the CLBF-MPC optimization problem of Eq. 7.22 for all states $x(t) \in \mathscr{U}_\rho$. The second part includes the proof of simultaneous closed-loop stability and process operational safety of the nonlinear system of Eq. 7.1 under the CLBF-MPC that uses an ensemble of RNN models of Eq. 2.4 for prediction.

Part 1: A feasible solution to the the CLBF-MPC optimization problem of Eq. 7.22 exists

for all times since it has been demonstrated in Propositions 7.2, 7.3 and 7.4 that the controllers $u = \Phi_{nn}(x) \in U$, $\forall x \in \mathscr{U}_\rho \backslash \mathscr{B}_\delta(x_e)$ and $u = \bar{u}(x) \in U$, $\forall x \in \mathscr{B}_\delta(x_e)$ in a sample-and-hold fashion satisfy the CLBF-MPC constraints of Eqs. 7.22d-7.22g. Specifically, the input constraint of Eq. 7.22d is satisfied over the prediction horizon since both $u = \Phi_{nn}(x)$ and $u = \bar{u}(x)$ are constrained by $u \in U$. The satisfaction of Eq. 7.22e is readily shown by letting $u(t_k) = \Phi_{nn}(x(t_k))$ when $x(t_k) \in \mathscr{U}_\rho \backslash (\mathscr{B}_\delta(x_e) \cup \mathscr{U}_{\rho_{nn}})$. Additionally, the input trajectories $u(t) = \Phi_{nn}(x(t_{k+i})) \in U$, $\forall t \in [t_{k+i}, t_{k+i+1})$ with $i = 0, ..., N-1$ is a set of feasible control actions that meet the constraint of Eq. 7.22f. In Proposition 7.4, it is shown that once the state is driven into $\mathscr{U}_{\rho_s}$ under the controller $u = \Phi_{nn}(x) \in U$, it will not leave $\mathscr{U}_{\rho_{nn}}$ within one sampling period for any $u \in U$. Therefore, the constraint of Eq. 7.22f is satisfied under the sample-and-hold implementation of $u = \Phi_{nn}(x) \in U$. Lastly, if $x(t_k) \in \mathscr{B}_\delta(x_e)$, $u(t) = \bar{u}(x(t_{k+i})) \in U$, $\forall t \in [t_{k+i}, t_{k+i+1})$ with $i = 0, ..., N-1$ is a set of control actions that meet the constraint of Eq. 7.22g as the controller $u = \bar{u}(x) \in U$ is designed to satisfy Eq. 7.15 to drive the state away from saddle points. This completes the proof of recursive feasibility for the CLBF-MPC optimization problem of Eq. 7.22.

Part 2: We first consider the case of an unbounded unsafe region $\mathscr{D}_u$ for the nonlinear system of Eq. 7.1. As there is no saddle point in the presence of $\mathscr{D}_u$ (i.e., $\mathbb{X}_e = \emptyset$), the last constraint of Eq. 7.22g in the CLBF-MPC optimization problem is never activated. Therefore, for any initial condition $x_0 \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_{nn}}$, the constraint of Eq. 7.22e forces the state to move towards the origin and drives the state into $\mathscr{U}_{\rho_{nn}}$ within finite sampling steps. After the state enters $\mathscr{U}_{\rho_{nn}}$, the constraint of Eq. 7.22f ensures boundedness of the state in $\mathscr{U}_{\rho_{nn}}$ for the remaining time. As a result, the nonlinear system of Eq. 7.1 is considered practically stable because it has been shown in Proposition 7.4 that the state of the nonlinear system of Eq. 7.1 is ultimately bounded in $\mathscr{U}_{\rho_{min}}$ (a small neighborhood around the origin). Additionally, it should be noted that since the state is also bounded in the safe operating region $\mathscr{U}_\rho$ for all times, which does not intersect with the unbounded unsafe region $\mathscr{D}_u$ in state-space, process operational safety for the system of Eq. 7.1 is guaranteed under CLBF-MPC.

Following the above analysis, the proof of closed-loop stability and process operational safety for a bounded unsafe region $\mathscr{D}_b$ is presented by showing that the state can be ultimately bounded

in $\mathscr{U}_{\rho_{nn}}$ instead of converging to saddle points. Starting from an initial condition $x_0 \in \mathscr{U}_{\rho} \backslash \mathscr{U}_{\rho_{nn}}$, the constraint of Eq. 7.22e drives the state towards the origin. However, the state may settle in saddle points (local minima of the CLBF) along its trajectory towards the origin if no further action is taken around the saddle points. To address this, the constraint of Eq. 7.22g is activated when $x(t_k) \in \mathscr{B}_{\delta}(x_e)$ to move the state away from the saddle points in a direction of decreasing the value of $W_c(x)$ such that the state can escape from saddle points and ultimately converge to the origin. Once the state leaves the neighborhood $\mathscr{B}_{\delta}(x_e)$ around the saddle points, closed-loop stability and process operation safety are still guaranteed under the constraints of Eqs. 7.22e-7.22f in the sense that the state of the nonlinear system of Eq. 7.1 stays in the safe operating region $\mathscr{U}_{\rho}$ for all times, and is ultimately maintained in $\mathscr{U}_{\rho_{nn}}$. This completes the proof of simultaneous closed-loop stability and operational safety for both an unbounded unsafe region and a bounded unsafe region. □

7.2.3 Online Learning of RNNs

Now we consider the nonlinear system of Eq. 7.1 subject to bounded time-varying disturbances (i.e.,$|w(t)| \leq w_M$, where w_M is greater than the sufficiently small bound w_m in Eq. 7.1) that cannot be fully eliminated by the sample-and-hold implementation of CLBF-based predictive controllers using the RNN models that are developed for the nominal system of Eq. 7.1 (i.e., $w(t) \equiv 0$). In this case, it is readily shown that the closed-loop system of Eq. 7.1 may be rendered unstable under the CLBF-based predictive controllers using the nominal RNN model (i.e., the RNN model obtained from open-loop simulations of the nominal system of Eq. 7.1 with $w(t) \equiv 0$) for all times since the modeling error between the nominal RNN model and the uncertain system of Eq. 7.1 no longer satisfies the constraint $|v| = |F(x,u,w) - F_{nn}(x,u)| \leq \gamma|x| \leq v_m$.

To account for the impact of disturbances in the predictions of the CLBF-MPC of Eq. 7.22, the RNN models of Eq. 7.22b need to be updated via online learning using the most recent process data to capture the nonlinear dynamics of the system of Eq. 7.1 subject to the time-varying disturbances $w(t)$. Event-triggered and error-triggered mechanisms can be utilized to implement online learning

of RNN models, e.g., [178, 179]. Specifically, the event-triggered mechanism updates the RNN model if the following inequality is violated for any $x \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_w}$:

$$W_c(x(t)) \leq W_c(x(t_k)) - \varepsilon_w(t - t_k),\ t \in [t_k, t_{k+1}) \tag{7.23}$$

where $\varepsilon_w > 0$. $\mathscr{U}_{\rho_w}$ with $\rho_w < \rho$ is characterized to be the largest level set of $W_c(x)$ within $\mathscr{U}_\rho$ such that if the current state is in $\mathscr{U}_{\rho_w}$, the value of $W_c(x)$ does not increase under the stabilizing controller $u = \Phi_{nn}(x)$ within one sampling period in the presence of bounded disturbances $|w(t)| \leq w_M$, i.e., $W_c(x(t)) < W_c(x(t_k)), \forall t \in [t_k, t_{k+1})$. Additionally, it also ensures that the closed-loop state is bounded in $\mathscr{U}_\rho$ and ultimately enters $\mathscr{U}_{\rho_w}$ for any initial state in $\mathscr{U}_\rho$. From Eq. 7.23, it is shown that the event-triggered mechanism activates the online update of RNN models if the decreasing rate of CLBF $W_c(x)$ is not satisfied within one sampling period. As a result, the RNN prediction accuracy is improved once the online learning is activated using the most recent process data and the closed-loop state can be driven into $\mathscr{U}_{\rho_w}$ at a faster rate.

In addition to the event-triggered mechanism, the following moving horizon error metric $E_{rnn}(t_k)$ is developed to indicate the RNN model prediction accuracy at $t = t_k$:

$$E_{rnn}(t_k) = \sum_{i=0}^{N_b} \frac{|x_p(t_{k-i}) - x(t_{k-i})|}{|x(t_{k-i})| + \delta} \tag{7.24}$$

where N_b is the number of sampling periods before t_k that contribute to the quantification of the prediction error. $x_p(t_{k-i})$ and $x(t_{k-i})$, $i = 0, ..., N_b$ are the predictions of the past states using RNN models, and the past state measurements from the actual nonlinear system of Eq. 7.1 under the same control actions, respectively. δ is a small positive real number that is introduced in the denominator of Eq. 7.24 to avoid the division by small numbers when $x(t_{k-i})$ approaches zero. The RNN model of Eq. 7.22b is updated if the accumulated error $E_{rnn}(t_k)$ exceeds the threshold E_T:

$$E_{rnn}(t_k) > E_T \tag{7.25}$$

where E_T is determined via extensive closed-loop simulations. It should be noted that when an online learning of RNN models is activated, all the data points since the last model update will be used as the training and validation data for the new RNN model. As the number of available data points has a great impact on the RNN model accuracy, N_b and E_T need to be carefully chosen to achieve a desired training performance. Specifically, the moving horizon length N_b is first determined via extensive closed-loop simulations to ensure that there are enough data points that can be utilized in the online update of RNN models, and meanwhile, will not cause data-storage burden. Subsequently, the threshold E_T is determined via simulations off-line to trigger an RNN model update when the state error has accumulated to an undesired level while accounting for common measurement noise, which is sufficiently small compared to time-varying disturbances from model uncertainty, and should not trigger an update of RNN models in most times. Additionally, when the state approaches the unsafe region, the threshold E_T should be adjusted to update online learning more frequently such that the new RNN models are able to capture the most recent dynamics subject to disturbances in a timely manner, and therefore, provide a sufficiently accurate prediction for the CLBF-MPC of Eq. 7.22 to avoid the unsafe region. Lastly, after the RNN model is updated at a certain sampling step $t = t_k$, all the errors before $t = t_k$ are reset to zero.

Remark 7.3. *It is noted that the event-triggered mechanism or the error-triggered mechanism could be activated when the conditions of Eq. 7.23 is violated, or the prediction error of Eq. 7.25 exceeds its threshold at a time instant $t = r_k$ that is within one sampling period, i.e., $r_k \in [t_k, t_{k+1})$. However, since the CLBF-MPC of Eq. 7.22 are implemented in a sample-and-hold fashion where the control actions remain the same for each sampling period Δ, i.e., $u = u(t_k)$, $\forall t \in [t_k, t_{k+1})$, the control actions will not be updated immediately after the RNN model update is triggered within one sampling period. In other words, if the online update of RNN models is triggered at $t = r_k \in [t_k, t_{k+1})$, the control actions will still be calculated at the next sampling time, i.e., $t = t_{k+1}$, using the updated RNN models. The asynchronization between the online learning of RNN models and the calculation of control actions using the new RNN models ensures that the*

sample-and-hold implementations of the CLBF-MPC of Eq. 7.22 remain unchanged, and also leaves enough computation time for RNN models to be updated using the most recent process data.

Remark 7.4. *The main objective of triggered model update is to improve the prediction accuracy of RNN models such that they are able to capture the most recent process dynamics subject to time-varying disturbances. Since the event-triggered mechanism updates RNN models only if the condition is violated, it is demonstrated that the event-triggered mechanism updates RNN models less frequently, and therefore, achieves better approximation performance due to more data available than the regular model update that is triggered every sampling period. Additionally, the frequency of online update depends on the threshold E_T. As a result, the optimal value of E_T is determined via extensive closed-loop simulations to achieve the desired closed-loop performance under disturbances.*

7.2.3.1 Implementation strategy for online RNN learning within CLBF-MPC

Based on the event-triggered and the error-triggered schemes, the implementation of online RNN learning is integrated with the machine-learning-based CLBF-MPC of Eq. 7.22 as follows:

Step 1 : An initial RNN model that will be utilized in the CLBF-MPC of Eq. 7.22 is derived from extensive open-loop simulations for the nominal system of Eq. 7.1 (i.e., $w(t) \equiv 0$) following the construction method in Section 2.2.2.

Step 2 : As shown in Fig. 7.1, starting from an initial condition $x_0 \in \mathscr{U}_\rho$, the nonlinear system of Eq. 7.1 is operated under CLBF-MPC in a sample-and-hold fashion with states being continuously monitored and collected. The online update of RNN models is triggered the moment that the decreasing rate of CLBF $W_c(x)$ of Eq. 7.23 is violated for any $x(t) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_w}$, or the moving horizon error detector of Eq. 7.24 exceeds its threshold E_T for any $x \in \mathscr{U}_{\rho_w} \backslash \mathscr{U}_{\rho_{min}}$. At the next sampling time, the new RNN model will replace the old model in the CLBF-MPC of Eq. 7.22 to solve for the optimal control actions $u^*(t)$ for the next sampling period.

Step 3 : When the closed-loop state enters a small neighborhood $\mathscr{U}_{\rho_{min}}$ around the origin, which is considered to be practically stable for the nominal system of Eq. 7.1, the error-triggering

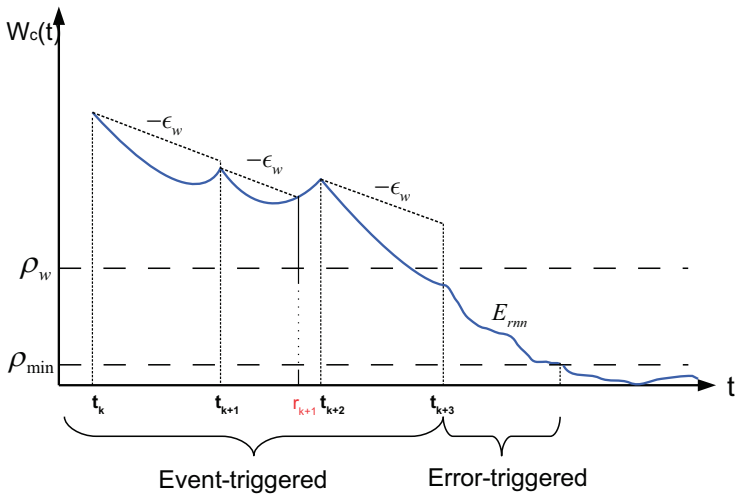


Figure 7.1: Evolution of CLBF $W_c(x)$ (blue trajectory) under the CLBF-MPC of Eq. 7.22 with event-triggered condition of Eq. 7.23 and error-triggered condition of Eq. 7.25, where the dashed lines with the slope $-\varepsilon_w$ represent the threshold lines in Eq. 7.23.

mechanism is taken off-line until the state leaves $\mathscr{U}_{\rho_{min}}$ again due to time-varying disturbances.

Remark 7.5. *It is noted that the online learning of RNN models is performed using the most recent process data only by loading the old RNN models with the previous RNN structure and weight matrices as initialization. Therefore, the new RNN models that are trained using new data points inherit some important features of the nominal process from the old RNN models and also capture the recent dynamics subject to time-varying disturbances from new data points. Additionally, instead of training a new RNN model from scratch, the training process based on the most recent data and the previous RNN model is more computationally tractable, and thus, can be readily incorporated in the real-time implementation of CLBF-MPC.*

Remark 7.6. *To ensure that there are enough data points for the online training of RNN models, an additional constraint for the number of collected data points can be employed with the event-triggered and the error-triggered mechanisms without affecting closed-loop stability or safety. Specifically, based on the definition of $\mathscr{U}_{\rho_w}$ in Eq. 7.23, the closed-loop state is guaranteed to move towards the origin every sampling period (maybe slowly) even if the online learning of RNN models is not activated. Therefore, it allows us to collect enough data points from multiple sampling periods to achieve a better training performance while maintaining the state in the*

closed-loop stability region. Additionally, in the error-triggered mechanism of Eq. 7.25, the moving horizon window length N_b for the prediction error of Eq. 7.24 needs to be carefully chosen to obtain a sufficient number of data points that will be utilized in the online update of RNN models.

7.2.4 Application to a Chemical Process Example

In this section, a chemical process example is utilized to illustrate the application of the proposed machine-learning based CLBF-MPC scheme to nonlinear systems with a bounded/unbounded unsafe region. We consider a well-mixed, non-isothermal continuous stirred tank reactor (CSTR) where an irreversible second-order exothermic reaction takes place. The reaction converts the reactant A to the product B via the chemical reaction $A \rightarrow B$. A heating jacket that supplies or removes heat from the reactor is used. The CSTR dynamic model derived from material and energy balances is given below:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 \tag{7.26a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{7.26b}$$

where C_A is the concentration of reactant A in the reactor, V is the volume of the reacting liquid in the reactor, T is the temperature of the reactor and Q denotes the heat input rate. The concentration of reactant A in the feed is C_{A0}. The feed temperature and volumetric flow rate are T_0 and F, respectively. The reacting liquid has a constant density of ρ_L and a heat capacity of C_p. ΔH, k_0, E, and R represent the enthalpy of reaction, pre-exponential constant, activation energy, and ideal gas constant, respectively. Process parameter values are listed in Table 7.1. The CSTR is initially operated at the unstable steady-state $(C_{As},\ T_s) = (1.95\ kmol/m^3,\ 402\ K)$, and $(C_{A0_s}\ Q_s) = (4\ kmol/m^3,\ 0\ kJ/hr)$. The manipulated inputs are the inlet concentration of species A and the heat input rate, which are represented by the deviation variables $\Delta C_{A0} = C_{A0} - C_{A0_s}$, $\Delta Q = Q - Q_s$, respectively. The manipulated inputs are bounded as follows: $|\Delta C_{A0}| \leq 3.5\ kmol/m^3$ and $|\Delta Q| \leq 5 \times 10^5\ kJ/hr$. The states and the inputs of the closed-loop system are $x^T = [C_A - C_{As}\ T - T_s]$

Table 7.1: Parameter values of the CSTR system.

$T_0 = 300\ K$	$F = 5\ m^3/hr$
$V = 1\ m^3$	$E = 5 \times 10^4\ kJ/kmol$
$k_0 = 8.46 \times 10^6\ m^3/kmol\ hr$	$\Delta H = -1.15 \times 10^4\ kJ/kmol$
$C_p = 0.231\ kJ/kg\ K$	$R = 8.314\ kJ/kmol\ K$
$\rho_L = 1000\ kg/m^3$	$C_{A0_s} = 4\ kmol/m^3$
$Q_s = 0.0\ kJ/hr$	

and $u^T = [\Delta C_{A0}\ \Delta Q]$, respectively, such that the equilibrium point of the system is at the origin of the state-space, (i.e., $(x_s^*, u_s^*) = (0,0)$). The explicit Euler method with an integration time step of $h_c = 2 \times 10^{-5}\ hr$ is applied to numerically simulate the dynamic model of Eq. 7.26. The nonlinear optimization problem of the CLBF-MPC of Eq. 7.22 is solved using the python module of the IPOPT software package [158], named PyIpopt with the sampling period $\Delta = 2 \times 10^{-3}\ hr$.

7.2.4.1 Development of RNN models

To develop an ensemble of RNN models that will be used in CLBF-MPC, extensive open-loop simulations are performed within the operating region for the CSTR of Eq. 7.26 to generate the dataset. Specifically, we run open-loop simulations with various initial states in state-space and inputs $u \in U$ for finite sampling steps such that the dataset is sufficiently large to be representative in the operating region. The sampled data points including states x and inputs u are saved with a minimum time step as the integration time step h_c. The RNN model is constructed with one input layer, two hidden layers consisting of 96 and 64 recurrent units, respectively, and one output layer. The inputs to the RNN model of Eq. 2.4 are the states $x(t_k)$ and the control actions $u(t_k)$ at $t = t_k$, $k = 0, 1, ...$, and the outputs are the predicted state trajectory over one sampling period (i.e., $t \in [t_k, t_{k+1}]$), where the data points with the time interval of h_c (i.e., the integration time step for the explicit Euler method) are treated as the internal states for RNN models. The sigmoid function is used as the activation function for RNN models, and early stopping is employed to avoid over-fitting. Additionally, we utilize a 10-fold cross validation to derive an ensemble of 10 RNN

models for the CLBF-MPC of Eq. 7.22. The optimal number of recurrent neural network models in the ensemble generally depends on the complexity of process dynamics and the size of datasets. In this example, the optimal number is determined by closed-loop simulations. Specifically, to determine the optimal number of neural network models, we first derive k RNN models based on a k-fold cross-validation. Subsequently, we start with a single RNN model and keep increasing the number of models used in MPC. The optimal number is determined to be the one when no further improvement of closed-loop performance is noticed for the increase of RNN models being used.

We first carry out open-loop simulation using the RNN model and the first-principles model of the CSTR system of Eq. 7.26, respectively. It should be noted that the machine learning approach is used when only data are available. The first-principles model in this study substitutes for the role of the experimental/industrial process. In other words, the simulation using first-principles model only serves as a benchmark to determine the best performance that any data-driven modeling method can achieve. In Chapter 2, we have demonstrated that starting from the same initial condition $x_0 \in \Omega_{\hat{\rho}}$ with the same input sequences, the state trajectories for a fixed finite interval of time under the RNN model are close to those under the first-principles model of the nonlinear CSTR of Eq. 7.26 (see Fig. 2.9). This implies that the RNN model can be regarded as a good representation for the CSTR first-principles model of Eq. 7.26 within the operating region.

7.2.4.2 Closed-loop simulation results

The control objective is to operate the CSTR at the unstable equilibrium point $(C_{As},\ T_s)$ and avoid the unsafe operating region (bounded and unbounded) in state-space by manipulating the heat input rate ΔQ and the inlet concentration ΔC_{A0} under the RNN-based CLBF-MPC. We first demonstrate the application of the proposed CLBF-MPC control scheme to an unbounded unsafe region $\mathscr{D}_u$ in state-space. The unsafe region is characterized as an unbounded set with high temperature and concentration for the CSTR of Eq. 7.26: $\mathscr{D}_u := \{x \in \mathbf{R}^2 \mid F(x) = x_1 + x_2 > 47\}$. It is noted that with the form of $F(x) = x_1 + x_2$, the temperature in the reactor is considered the dominant factor in characterizing the unsafe region $\mathscr{D}_u$, while the reactant concentration is also accounted for because

of its impact on the reaction rate $r = k_0 e^{-E/RT} C_A^2$. Following the construction method of a CLBF in Section 6.4.2, we first design a control Lyapunov function with the standard quadratic form $V(x) = x^T P x$, where P is a positive definite matrix as follows:

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{7.27}$$

Then, we characterize a set H that contains $\mathscr{D}_u$: $H := \{x \in \mathbf{R}^2 \mid F(x) > 45\}$, and design the control barrier function $B(x)$ as follows:

$$B(x) = \begin{cases} e^{F(x)-47} - 2 \times e^{-2}, & \text{if } x \in H \\ -e^{-2}, & \text{if } x \notin H \end{cases} \tag{7.28}$$

The control Lyapunov-barrier function $W_c(x) = V(x) + \mu B(x) + \nu$ is constructed with the following parameters: $\hat{\rho} = 0$, $c_1 = 0.1$, $c_2 = 1061$, $c_3 = 5808$, $c_4 = 2259$, $\nu = \hat{\rho} - c_1 c_4 = -225.9$, and $\mu = 4.6 \times 10^7$. It is demonstrated in Fig. 7.2 that under the CLBF-MPC of Eq. 7.22, all the trajectories starting from initial states in $\mathscr{U}_{\hat{\rho}}$ (a subset of the safe operating region $\mathscr{U}_\rho$ in state-space) avoid the unbounded unsafe region $\mathscr{D}_u$ on the top and converge to $\mathscr{U}_{\rho_{min}}$.

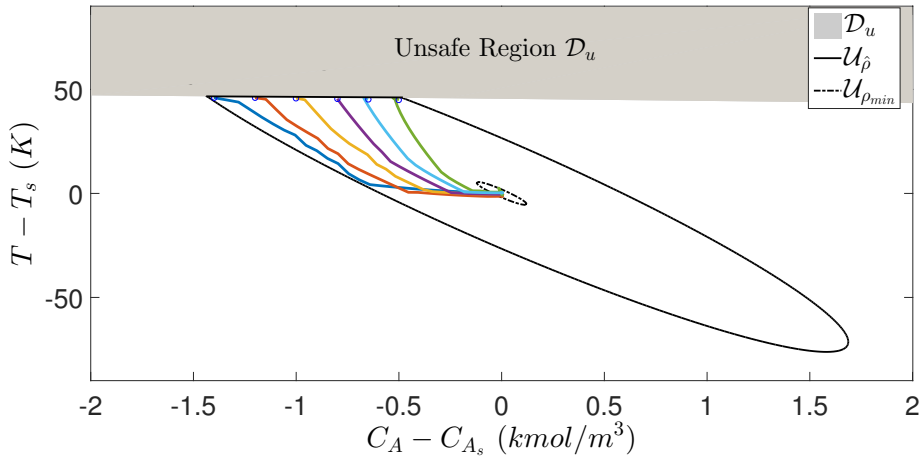


Figure 7.2: Closed-loop state trajectories for the CSTR system of Eq. 7.26 under the CLBF-MPC using an ensemble of RNN models. The initial conditions are marked by circles, and the set of unbounded unsafe states $\mathscr{D}_u$ is the gray area on the top.

The second example is used to demonstrate that the state of the closed-loop system of Eq. 7.26 can avoid a bounded unsafe region $\mathscr{D}_b$ in state-space and converge to a small neighborhood around the origin under the CLBF-MPC of Eq. 7.22. To demonstrate that the state is able to pass around the unsafe region along the trajectory towards the origin, we design a bounded unsafe region $\mathscr{D}_b$ embedded within the safe operating region as shown in the above example. Specifically, the unsafe region is defined as an ellipse: $\mathscr{D}_b := \{x \in \mathbf{R}^2 \mid F(x) = \frac{(x_1+0.92)^2}{1} + \frac{(x_2-42)^2}{500} < 0.06\}$. H is defined as $H := \{x \in \mathbf{R}^2 \mid F(x) < 0.07\}$. The control barrier function $B(x)$ is defined as follows:

$$B(x) = \begin{cases} e^{\frac{F(x)}{F(x)-0.07}} - e^{-6}, & \text{if } x \in H \\ -e^{-6}, & \text{if } x \notin H \end{cases} \tag{7.29}$$

Using the same control Lyapunov function $V(x)$ as in the first example, the control Lyapunov-barrier function $W_c(x) = V(x) + \mu B(x) + \nu$ is constructed with the following parameters: $\rho_c = 0, c_1 = 0.1, c_2 = 1061, c_3 = \max_{x \in \partial \mathscr{H}} |x|^2 = 2295, c_4 = \min_{x \in \partial \mathscr{D}} |x|^2 = 1370, \nu = \rho_c - c_1 c_4 = -160$. Hence, μ is chosen to be 1×10^9 to satisfy the construction rules in Section 6.4.2 and $\mathscr{U}_{\hat{\rho}}$ with $\hat{\rho} = -2.47 \times 10^6$ is the stability and safety region in the simulation. Additionally, we calculate the stationary point of $W_c(x)$ (other than the origin) in state-space by letting $\frac{\partial W_c(x)}{\partial x} = 0$. It is obtained that the stationary point is $x_e = (-1.004,\ 47.48)$ and it turns out to be a saddle point from partial derivative test (i.e., x_e is a saddle point if the determinant of the Hessian matrix of $W_c(x)$ at x_e is negative).

In Fig. 7.3, it is demonstrated that for all initial states x_0 in $\mathscr{U}_{\hat{\rho}}$ (marked by circles), the closed-loop trajectories avoid the bounded unsafe region $\mathscr{D}_b$ that is embedded within $\mathscr{U}_{\hat{\rho}}$, and ultimately converges to $\mathscr{U}_{\rho_{min}}$ under CLBF-MPC.

7.2.4.3 Comparison with a linear state-space model

Additionally, to demonstrate the merits of the machine-learning-based CLBF-MPC in terms of desired prediction accuracy and guaranteed process operational safety, a linear state-space model is derived using the same dataset for the RNN models to approximate the nonlinear dynamics in the

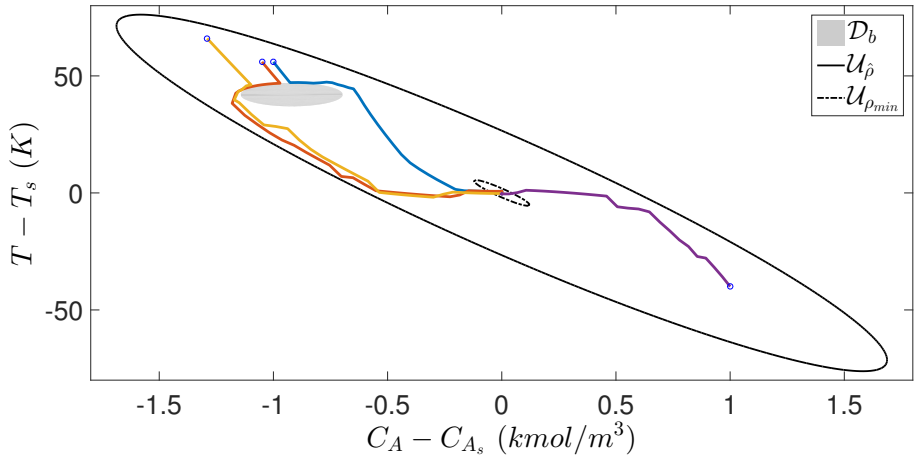


Figure 7.3: Closed-loop state trajectories for the system of Eq. 7.26 under the CLBF-MPC using an ensemble of RNN models. The initial conditions are marked by circles, and the set of bounded unsafe states $\mathscr{D}_b$ is the gray area embedded within $\mathscr{U}_{\hat{\rho}}$.

operating region for comparison. Specifically, the linear state-space model for the CSTR system of Eq. 7.26 is developed with the following form:

$$\dot{x} = A_s x + B_s u \tag{7.30}$$

where x and u are the state vector and the manipulated input vector, A_s and B_s are coefficient matrices for the state-space model. Following the system identification method in [74], the numerical algorithms for subspace state space system identification is utilized to obtain A_s and B_s as follows:

$$A_s = 100 \times \begin{bmatrix} -0.154 & -0.003 \\ 5.19 & 0.138 \end{bmatrix}, \quad B_s = \begin{bmatrix} 4.03 & 0 \\ 1.23 & 0.004 \end{bmatrix} \tag{7.31}$$

The eigenvalues of matrix A_s is calculated to be $\lambda_1 = -5$ and $\lambda_2 = 3.14$, which is consistent with the fact that the steady-state $(C_{As},\ T_s) = (1.95\ kmol/m^3,\ 402\ K)$ is an unstable equilibrium point of CSTR.

It is shown in Fig. 7.4 that for some initial conditions in $\mathscr{U}_{\hat{\rho}}$, the closed-loop state trajectories are able to avoid the unsafe region and converge to the steady-state under the MPC using a linear

model. However, in Fig. 7.5, it is demonstrated that for some other initial conditions, the state trajectories (with dashed line) enter the unsafe region due to a considerable model mismatch of the linear state-space model. It is noted that the model predictive controller using a simple linear state-space model is generally able to stabilize the nonlinear system in a neighborhood around the steady-state provided that the model mismatch between the linear model and the nonlinear system is small in the neighborhood. However, the MPC using a linear state-space model does not work in this example because in addition to closed-loop stability, we are addressing process operational safety that requires a sufficiently small model mismatch for which feedback control without an accurate process model cannot guarantee that the process state avoids the unsafe region for all times. In fact, in the presence of a large model mismatch, feedback control cannot prevent the state from entering the unsafe region since the state predicted by the linear model may be outside of the unsafe region while the true state actually enters it within one sampling period. Therefore, it motivates us to use an ensemble of RNN models with a sufficiently small model mismatch to approximate nonlinear dynamics in the operating region and provide sufficiently accurate predictions for MPC.

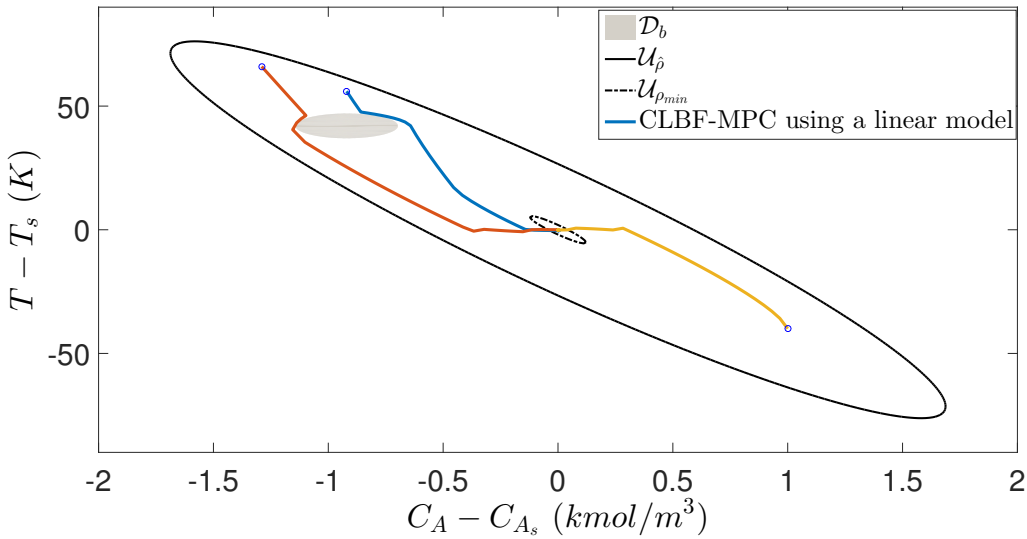


Figure 7.4: Closed-loop state trajectories for the CSTR system under the CLBF-MPC using a linear state-space model. The initial conditions are marked by circles, and the set of bounded unsafe states $\mathscr{D}_b$ is the gray ellipse in state-space.

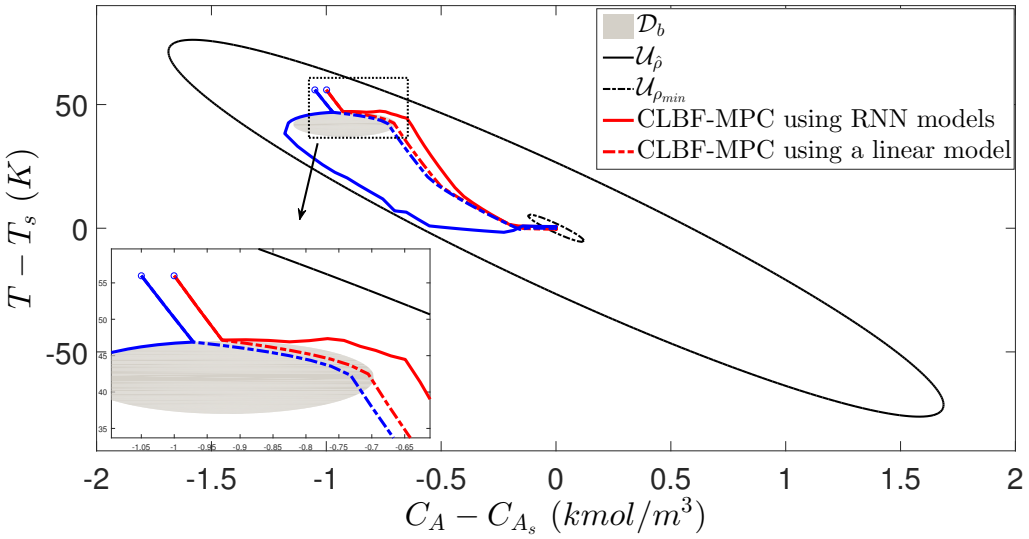


Figure 7.5: Comparison of the closed-loop state trajectories under the CLBF-MPC using a linear state-space model (dashed) and an ensemble of RNN models (solid), respectively. The initial conditions are marked by circles, and the set of bounded unsafe states $\mathscr{D}_b$ is the gray ellipse in state-space.

The above two case studies demonstrate that the RNN models that are developed from extensive open-loop simulations to replace the CSTR process of Eq. 7.26 in CLBF-MPC achieve a desired approximation performance. Subsequently, the CLBF-MPC of Eq. 7.22 based on an ensemble of RNN models guarantees that for any initial condition in the safe operating region, the state of the closed-loop system of Eq. 7.26 is maintained within the safe operating region for all times, and is able to converge to a small neighborhood $\mathscr{U}_{\rho_{min}}$ around the origin ultimately while avoiding the unsafe region (bounded and unbounded) in state-space.

7.2.4.4 Real-time CLBF-MPC with online learning of RNN models

Under the CLBF-MPC of Eq. 7.22, we consider the model variations due to the following disturbances: (1) the feed flow rate F is changing from 5 m^3/h to 7 m^3/h at $t = 0$ hr, and (2) the actual value of the pre-exponential constant k_0 used in the process model is reduced by half to represent a change in the reaction rate at the simulation time $t = 0$ hr. The closed-loop simulation results for the CSTR of Eq. 7.26 under the CLBF-MPC with and without online learning of RNN

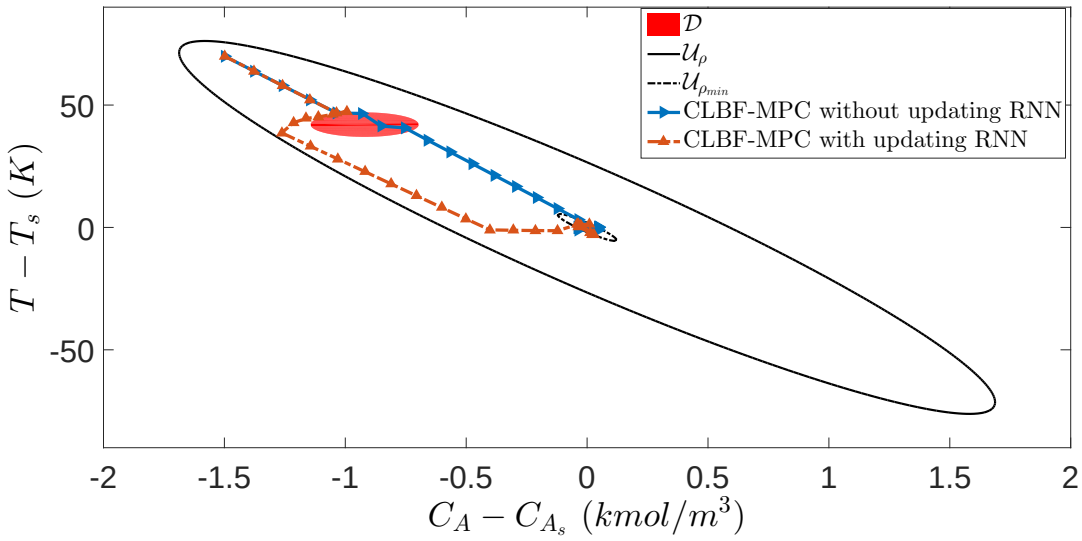


Figure 7.6: The state-space profiles for the closed-loop CSTR subject to time-varying disturbances under the CLBF-MPC of Eq. 7.22 with (red trajectory) and without online RNN update (blue trajectory), respectively, for an initial condition (-1.5,70).

models, respectively, are shown in Figs. 7.6-7.8. Specifically, in Fig. 7.6, it is demonstrated that in the presence of disturbances, the closed-loop state trajectory under the CLBF-MPC using online update of RNN models is able to avoid the unsafe region and converge to a small neighborhood around the origin, while the one under the CLBF-MPC without online RNN update crosses the red unsafe region $\mathscr{D}$ due to a considerable model mismatch between the initial RNN model for the nominal process of Eq. 7.26 and the actual process subject to disturbances. Fig. 7.7 shows the input profiles under the CLBF-MPC with and without online RNN update, from which recursive feasibility and satisfaction of input constraints are demonstrated for both optimization problems. Additionally, it is observed in Fig. 7.7 that since RNN models are updated in a timely manner under the CLBF-MPC with online learning, the oscillation of u_1 becomes less near the end of operation period compared to the one without online update.

In the closed-loop simulation, it is demonstrated that the event-triggered mechanism of Eq. 7.23 is not activated as the closed-loop state moves towards the origin quickly. Therefore, the value of the accumulated prediction errors $E_{rnn}(t)$ of Eq. 7.24 is shown in Fig. 7.8 for CLBF-MPCs with and without online RNN update, respectively, to show the real-time prediction accuracy of the

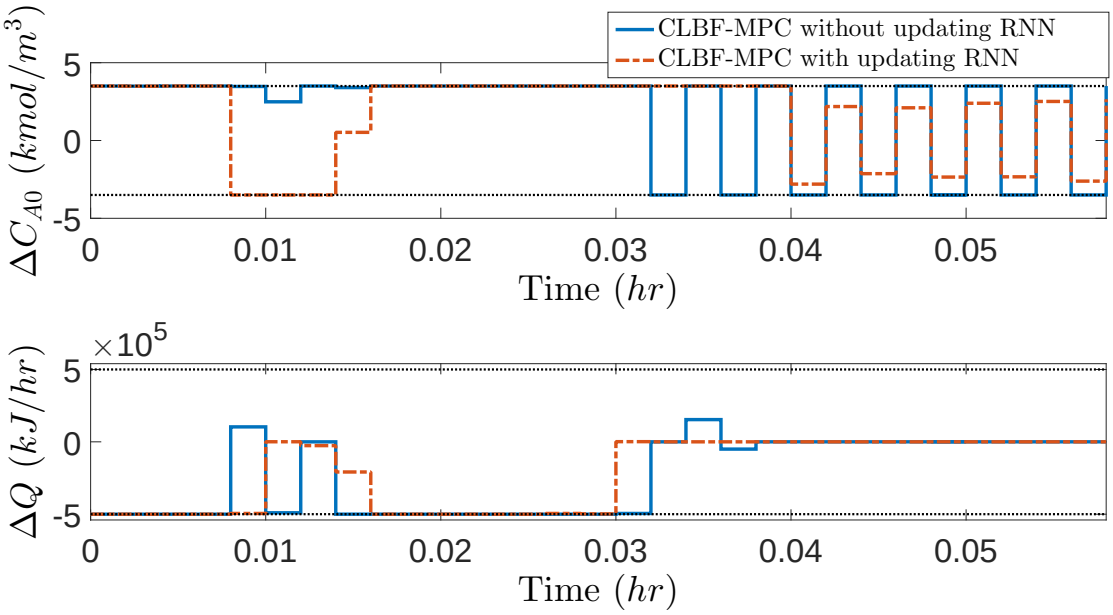


Figure 7.7: Manipulated input profiles ($u_1 = \Delta C_{A0}$, $u_2 = \Delta Q$) for the closed-loop CSTR subject to time-varying disturbances under the CLBF-MPC of Eq. 7.22 with (red profile) and without online RNN update (blue profile), respectively, for an initial condition (-1.5,70).

RNN models. Fig. 7.8 demonstrates that without online learning, the error (blue lines) exceeds the threshold (left y-axis) quickly and increases to an undesired level during the operation, which implies the failure of the initial RNN model in capturing the actual CSTR dynamics in the presence of disturbances. However, under the CLBF-MPC with online RNN learning, it is demonstrated that the RNN model update is triggered six times during the entire operation period (i.e., from t=0 *hr* to t=0.06 *hr*) to maintain the error (red lines) below its threshold (right y-axis) for most of the time. Therefore, by using online learning, the RNN models in CLBF-MPC always capture the latest process dynamics subject to disturbances, and lead to a desired closed-loop performance for the CSTR of Eq. 7.26 in terms of simultaneous closed-loop stability and operational safety.

7.3 CLBF-EMPC Using RNN models

To achieve higher economic profitability than the steady-state operation of the nonlinear system of Eq. 7.1, economic model predictive control scheme (EMPC) that is formulated with an economic

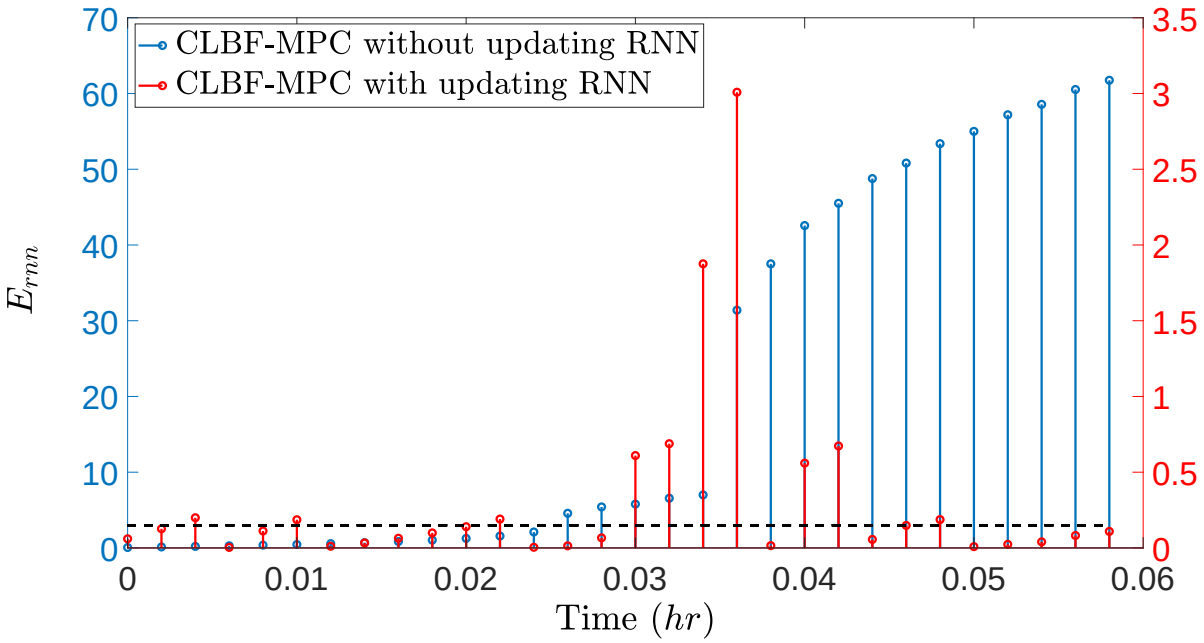


Figure 7.8: Value of $E_{rnn}(t)$ at each sampling time for the closed-loop CSTR subject to time-varying disturbances under the CLBF-MPC of Eq. 7.22 with (red, right y-axis) and without online RNN update (blue, left y-axis), respectively, where the threshold E_T is set to 0.15 (dashed horizontal line corresponding to the right y-axis).

objective function to operate the system in a time-varying fashion is utilized in this section. See, also, Section 6.6 for designs of EMPC accounting for operational safety. Specifically, based on the RNN-based CLBF-MPC of Eq. 7.22, an RNN-based economic model predictive controller with CLBF-based constraints (i.e., CLBF-EMPC) is developed in this section. Similarly, the ensemble learning of multiple RNN models are used to improve the overall prediction performance. k-fold cross-validation is used to train k distinct RNN models for the same process, (i.e., the nonlinear system of Eq. 7.1), and the final prediction results are obtained by taking average of k RNN predictions. Based on the ensemble of RNN models, the CLBF-EMPC scheme using RNN models

is represented by the following optimization problem [173]:

$$\max_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(t), u(t))dt \tag{7.32a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = \frac{1}{N_e}\sum_{j=1}^{N_e} F_{nn}^{j}(\tilde{x}(t), u(t)) \tag{7.32b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{7.32c}$$

$$u(t) \in U, \ \forall\, t \in [t_k, t_{k+N}) \tag{7.32d}$$

$$W_c(\tilde{x}(t)) \leq \rho_e, \ \forall\, t \in [t_k, t_{k+N}), \ \text{ if } W_c(x(t_k)) \leq \rho_e \tag{7.32e}$$

$$\dot{W}_c(x(t_k), u(t_k)) \leq \dot{W}_c(x(t_k), \Phi_{nn}(t_k)), \ \text{ if } W_c(x(t_k)) > \rho_e \tag{7.32f}$$

where the notation follows that in Eq. 7.22 and the CLBF-EMPC is again implemented in a sample-and-hold fashion. Unlike the CLBF-MPC objective function $l_t(x,u)$ of Eq. 7.22a that has its minimum value at the steady-state, the objective function $l_e(x,u)$ of Eq. 7.32a represents the process economic performance and will be maximized over the prediction horizon. Two CLBF-based constraints are incorporated in the design of CLBF-EMPC to ensure boundedness of the state in the safe stability region $\mathscr{U}_\rho$ in Eq. 7.4. The constraint of Eq. 7.32e is activated when the state $x(t_k)$ is in $\mathscr{U}_{\rho_e}$, where $\rho_e < \rho$. When the state leaves $\mathscr{U}_{\rho_e}$ due to disturbances or model mismatch (which will be discussed in the following section), the constraint of Eq. 7.32f is applied to drive the state towards the origin. As a result, the state will move into $\mathscr{U}_{\rho_e}$ within finite sampling periods. Additionally, we assume that the state measurements of the closed-loop system of Eq. 7.1 is available at each sampling time. The CLBF-EMPC optimization problem of Eq. 7.32 will calculate an optimal input sequence $u^*(t)$, $\forall t \in [t_k, t_{k+N})$, but only the first control action of $u^*(t)$ will be applied over the next sampling period.

7.3.1 Stability and Safety Under CLBF-EMPC

Closed-loop stability and safety for the nonlinear system of Eq. 7.1 under the CLBF-EMPC of Eq. 7.32 will be proven in this section. It should be noted that for the operation of the nonlinear

system of Eq. 7.1 under EMPC, the system is considered stable and safe if the state can be bounded in a safe stability region for all times for any initial condition inside of this region. In other words, the system is not required to be operated at the steady-state like what it is under CLBF-MPC since it is demonstrated that economic performance can be improved under time-varying operation than the steady-state operation. The following proposition is developed to demonstrate that the feedback controller $u = \Phi_{nn}(x) \in U$ that maintains the state of the RNN model of Eq. 2.4 in the safe operating region $\mathscr{U}_\rho$ also ensures the boundedness of the state of the nonlinear system of Eq. 7.1 within $\mathscr{U}_\rho$ accounting for bounded disturbances (i.e., $|w(t)| \leq w_m$), bounded modeling error (i.e., $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma|x| \leq \nu_m$) and sample-and-hold implementation of control actions.

Proposition 7.5. *Consider the system of Eq. 7.1 under the sample-and-hold implementation of the controller $u = \Phi_{nn}(x) \in U$ that meets the conditions of Eq. 7.5. If there exists a positive real number $\gamma < \hat{c}_3/\hat{c}_4$ such that for all $x \in \mathscr{U}_\rho$ and $u \in U$, the modeling error between the RNN model of Eq. 2.4 and the nonlinear system of Eq. 7.1 is constrained by $|\nu| = |F(x,u,0) - F_{nn}(x,u)| \leq \gamma|x|$, and there exist $\varepsilon_w > 0$, $\Delta > 0$ and $\rho > \rho_e$ that satisfy*

$$-\frac{\tilde{c}_3}{\hat{c}_2}(\rho_e - \rho_0) + L_x^{'} M\Delta + L_w^{'} w_m \leq -\varepsilon_w \tag{7.33a}$$

$$\rho_e \leq \rho - f_e(f_w(\Delta)) \tag{7.33b}$$

$$\mathbb{X}_e \subset \mathscr{U}_{\rho_e} \tag{7.33c}$$

where $f_w(t)$ and $f_e(t)$ are given by Eq. 7.7a and Eq. 7.15, respectively, then for any $x(t_k) \in \mathscr{U}_\rho$, the state of the nonlinear system of Eq. 7.1 is guaranteed to be bounded in $\mathscr{U}_\rho$ for all times.

Proof. We first prove that $\dot{W}_c(x)$ based on the state of the nonlinear system of Eq. 7.1 can be rendered negative under continuous implementation of $u = \Phi_{nn}(x) \in U$ for any $x \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$. The

time-derivative of $W_c(x)$, $\forall x \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$ is derived as follows using Eq. 7.5b and Eq. 7.5c:

$$\begin{aligned}
\dot{W}_c =& \frac{\partial W_c(x)}{\partial x} F(x, \Phi_{nn}(x), 0) \\
=& \frac{\partial W_c(x)}{\partial x} (F_{nn}(x, \Phi_{nn}(x)) + F(x, \Phi_{nn}(x), 0) - F_{nn}(x, \Phi_{nn}(x))) \\
\leq& -\hat{c}_3 |x|^2 + \hat{c}_4 |x| (F(x, \Phi_{nn}(x), 0) - F_{nn}(x, \Phi_{nn}(x))) \\
\leq& -\hat{c}_3 |x|^2 + \hat{c}_4 \gamma |x|^2
\end{aligned} \tag{7.34}$$

Therefore, if γ is constrained by $\gamma < \hat{c}_3/\hat{c}_4$, it holds that $\dot{W}_c \leq -\tilde{c}_3 |x|^2 < 0$, $\forall x \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$ by letting $\tilde{c}_3 = -\hat{c}_3 + \hat{c}_4 \gamma$. Next, we consider the impacts of bounded disturbances and of the sample-and-hold implementation of control actions (i.e., $u(t) = u(t_k)$, $\forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$ and Δ is the sampling period) on closed-loop stability of the nonlinear system of Eq. 7.1. Assuming $x(t_k) = \hat{x}(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_s}$, the time-derivative of $W_c(x)$ in Eq. 7.34 for the nonlinear system of Eq. 7.1 subject to bounded disturbances (i.e., $|w| \leq w_m$) can be derived as follows:

$$\begin{aligned}
\dot{W}_c(x(t)) =& \frac{\partial W_c(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\
=& \frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) + \frac{\partial W_c(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\
& - \frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0)
\end{aligned} \tag{7.35}$$

Using Eq. 7.5b, Eq. 7.34 and the Lipschitz condition in Eq. 7.6, $\dot{W}_c(x(t))$ is bounded by the the following inequality for all $t \in [t_k, t_{k+1})$ and $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$:

$$\begin{aligned}
\dot{W}_c(x(t)) \leq& -\frac{\tilde{c}_3}{\hat{c}_2} (\rho_e - \rho_0) + \frac{\partial W_c(x(t))}{\partial x} F(x(t), \Phi_{nn}(x(t_k)), w) \\
& - \frac{\partial W_c(x(t_k))}{\partial x} F(x(t_k), \Phi_{nn}(x(t_k)), 0) \\
\leq& -\frac{\tilde{c}_3}{\hat{c}_2} (\rho_e - \rho_0) + L_x' |x(t) - x(t_k)| + L_w' |w| \\
\leq& -\frac{\tilde{c}_3}{\hat{c}_2} (\rho_e - \rho_0) + L_x' M \Delta + L_w' w_m
\end{aligned} \tag{7.36}$$

From Eq. 7.36, it is obtained that $\dot{W}_c(x(t)) \leq -\varepsilon_w$ holds for all $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$ and $t \in [t_k, t_{k+1})$ if

Eq. 7.33a is satisfied.

So far we have demonstrated that for any state $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, the state does not leave $\mathscr{U}_\rho$ under the the sample-and-hold implementation of $u = \Phi_{nn}(x) \in U$. It remains to show that for $x(t_k) \in \mathscr{U}_{\rho_e}$, the state of the nonlinear system of Eq. 7.1 will not leave $\mathscr{U}_\rho$ within one sampling period if the state predicted by the RNN system of Eq. 2.4 is bounded in $\mathscr{U}_{\rho_e}$. Specifically, for any $x(t_k) = \hat{x}(t_k) \in \mathscr{U}_{\rho_e}$, the following inequality is derived based on Eq. 7.7 and Eq. 7.15 for $t \in [t_k, t_{k+1})$:

$$\begin{aligned} W_c(x(t)) &\leq W_c(\hat{x}(t)) + f_e(|x(t) - \hat{x}(t)|) \\ &\leq W_c(\hat{x}(t)) + f_e(f_w(t - t_k)) \\ &\leq \rho_e + f_e(f_w(\Delta)) \end{aligned} \tag{7.37}$$

Therefore, if $\mathscr{U}_{\rho_e}$ is characterized to satisfy Eq. 7.33b, it follows that $W_c(x(t)) \leq \rho$, which implies that the state of the nonlinear system of Eq. 7.1 is bounded in $\mathscr{U}_\rho$ within one sampling period. This completes the proof that the closed-loop state of the nonlinear system of Eq. 7.1 subject to bounded disturbances (i.e., $|w(t)| \leq w_m$) is guaranteed to be bounded in the safe operating region $\mathscr{U}_\rho$ for any initial condition x_0 in this region under the sample-and-hold implementation of $u = \Phi_{nn}(x) \in U$. □

Remark 7.7. *The issue of convergence to the stationary points (for $x \neq 0$) in the presence of a bounded unsafe region $\mathscr{D}_b$ is addressed by designing the set $\mathscr{U}_{\rho_e}$ to include the set of stationary points inside (i.e., Eq. 7.33c). Specifically, since the state of the nonlinear system of Eq. 7.1 is not required to move towards the origin (or any stationary points) within $\mathscr{U}_{\rho_e}$ under the constraint of Eq. 7.32e, the state will not get stuck in a stationary point unless it is exactly the state where the objective function of CLBF-EMPC of Eq. 7.32 attains its maximum value. Therefore, boundedness of the state in $\mathscr{U}_\rho$ is guaranteed for the nonlinear system of Eq. 7.1 with both bounded and unbounded unsafe regions when implementing $u = \Phi_{nn}(x) \in U$ in a sample-and-hold fashion. However, it should be noted that when the system is required to be operated at the steady-state under a tracking MPC, e.g., CLBF-MPC, Theorem 7.2 has been established to show that the stationary points need to be handled by a set of discontinuous control actions such that the state*

can escape from the stationary points and converge to the origin.

Based on Propositions 7.1 and 7.5, we establish the following theorem to demonstrate closed-loop stability and process operational safety guarantees for the nonlinear system of Eq. 7.1 under the CLBF-EMPC of Eq. 7.32.

Theorem 7.3. *Consider the system of Eq. 7.1 with a CLBF W_c that satisfies Eq. 7.4. If there exist $\rho > \rho_e$ and $\gamma < \hat{c}_3/\hat{c}_4$ that satisfy the conditions in Propositions 7.1 and 7.5, then given any initial state $x_0 \in \mathscr{U}_\rho$, recursive feasibility of the CLBF-EMPC optimization problem of Eq. 7.32 and boundedness of the state in the safe stability region $\mathscr{U}_\rho$ are guaranteed for all times.*

Proof. We first prove that a set of feasible solution exists for the CLBF-EMPC optimization problem of Eq. 7.32 for all states $x(t) \in \mathscr{U}_\rho$ by showing that the input trajectories $u(t) = \Phi_{nn}(x(t_{k+i})) \in U$, $\forall t \in [t_{k+i}, t_{k+i+1})$ with $i = 0, ..., N-1$ meet the constraints of the CLBF-EMPC optimization problem of Eq. 7.32. The discussion mainly focuses on the constraints of Eqs. 7.32e-7.32f as the satisfaction of the input constraint $u \in U$ of Eq. 7.32d is readily shown for the controller $u = \Phi_{nn}(x) \in U$. Specifically, if $x(t_k) \in \mathscr{U}_{\rho_e}$, the constraint of Eq. 7.32e is satisfied under the sample-and-hold implementation of $u = \Phi_{nn}(x) \in U$ since the state of the RNN system of Eq. 2.4 will be steered towards the origin or the stationary points in the presence of a bounded unsafe region. In any case, the state is maintained in $\mathscr{U}_{\rho_e}$ under $u = \Phi_{nn}(x) \in U$. On the other hand, if $x(t_k) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, the set of control actions $u(t) = \Phi_{nn}(x(t_{k+i})) \in U$, $i = 0, ..., N-1$ is again a feasible solution that meets the constraints of Eq. 7.32f (i.e., the inequality constraint of Eq. 7.32f becomes an active constraint). This completes the proof of recursive feasibility for the CLBF-EMPC of Eq. 7.32.

The proof of boundedness of the state in $\mathscr{U}_\rho$ follows the conclusions in Propositions 7.1 and 7.5. We first consider the case where $x(t_k) \in \mathscr{U}_{\rho_e}$. As it is required by the constraint of Eq. 7.32e that the state $x(t)$, $\forall t \in [t_k, t_{k+1})$ predicted by the ensemble of RNN models of Eq. 7.32b be bounded in $\mathscr{U}_{\rho_e}$, it follows from Eq. 7.37 that the state of the nonlinear system of Eq. 7.1 does not leave $\mathscr{U}_\rho$ within one sampling period. At the next sampling period, if $x(t_{k+1})$ remains in $\mathscr{U}_{\rho_e}$, it is again

bounded in $\mathscr{U}_\rho$ following the above analysis. However, if $x(t_{k+1})$ enters $\mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, the constraint of Eq. 7.32e is activated to drive the state of the RNN model of Eq. 2.4 towards the origin. Since it is proven in Proposition 7.5 that $\dot{W}_c$ based on the state of the nonlinear system of Eq. 7.1 can be rendered negative accounting for bounded disturbances and modeling error within one sampling period under $u = \Phi_{nn}(x) \in U$ (a feasible solution to the CLBF-EMPC optimization problem), the state of the nonlinear system of Eq. 7.1 is also able to move towards the origin and ultimately enters $\mathscr{U}_{\rho_e}$ within finite sampling periods. This completes the proof of closed-loop stability of the nonlinear system of Eq. 7.1 under CLBF-EMPC.

Additionally, since the safe stability region $\mathscr{U}_\rho$ does not intersect with the (bounded and unbounded) unsafe region (i.e., $\mathscr{U}_\rho \cap \mathscr{D} = \emptyset$) according to the definition of the constrained CLBF of Eq. 7.4, the state trajectory under the time-varying operation of the nonlinear system of Eq. 7.1 does not enter the unsafe region for all times. Therefore, process operational safety in terms of avoidance of the unsafe region is also guaranteed under CLBF-EMPC. □

7.3.1.1 Online learning of RNN models

Closed-loop stability and safety in Theorem 7.3 holds for the system of Eq. 7.1 subject to bounded disturbances (i.e., $|w(t)| \leq w_m$) as the effects of disturbances have been accounted for in the sample-and-hold implementation of the control actions, which requires the disturbances $w(t)$ and the sampling period Δ to be sufficiently small such that Eq. 7.36 is satisfied. However, in the presence of time-varying disturbances that are not sufficiently small, e.g., $|w(t)| \leq w_M$ where $w_M > w_m$, the nonlinear system of Eq. 7.1 may lose closed-loop stability and safety in terms of boundedness of the state in the safe stability region due to a considerable model mismatch between the actual nonlinear process under disturbances and the RNN models that are developed for the nominal system of Eq. 7.1 with $w(t) \equiv 0$. In this case, real-time adaptive machine-learning-based predictive control can be employed to mitigate the impact of disturbances by updating RNN models online using the most recent process data. Following the discussion of online learning of RNN models in Section 7.2.3, the implementation strategy of online update of RNN models within

CLBF-EMPC is given as follows:

Step 1 : An initial ensemble of RNN models for the nominal system of Eq. 7.1 (i.e., $w(t) \equiv 0$) are developed to approximate the nonlinear dynamics in the operating region $\mathscr{U}_\rho$.

Step 2 : The nonlinear system of Eq. 7.1 is operated under CLBF-EMPC in a sample-and-hold fashion and the states are continuously monitored and collected. For any $x(t) \in \mathscr{U}_\rho \backslash \mathscr{U}_{\rho_e}$, where $\mathscr{U}_{\rho_e} \subset \mathscr{U}_{\rho_w}$, the online learning of RNN models is activated following the event-triggered mechanism of Eq. 7.23. For any $x(t) \in \mathscr{U}_{\rho_e}$, the error-triggered mechanism of Eq. 7.25 is utilized to adapt the RNN models to the time-varying disturbances using the most recent process data. Similarly, at the next sampling time, the CLBF-EMPC of Eq. 7.32 will use the updated RNN model to calculate the optimal control actions $u^*(t)$ for the next sampling period.

7.3.2 Application to a Chemical Process Example

We consider the same chemical process example as in Section 7.2.4 to illustrate the application of CLBF-EMPC using an ensemble of RNN models. The dynamic process model of the continuous stirred tank reactor (CSTR) and the parameter values are given in Eq. 7.26 and Table 7.1, respectively, and are omitted here. The CSTR is initially operated at the unstable steady-state $(C_{As},\ T_s) = (1.95\ kmol/m^3,\ 402\ K)$, and $(C_{A0_s}\ Q_s) = (4\ kmol/m^3,\ 0\ kJ/hr)$. The states x and the manipulated inputs u of the closed-loop CSTR system are represented in deviation forms, i.e., $x^T = [C_A - C_{As}\ T - T_s]$ and $u^T = [\Delta C_{A0}\ \Delta Q]$, respectively. Additionally, the manipulated inputs are bounded as follows: $|\Delta C_{A0}| \leq 3.5\ kmol/m^3$ and $|\Delta Q| \leq 5 \times 10^5\ kJ/hr$. The explicit Euler method with an integration time step of $h_c = 2 \times 10^{-5}\ hr$ is applied to numerically simulate the dynamic model of Eq. 7.26. Additionally, the ensemble of RNN models are developed following the same approach as performed in Section 7.2.4.1.

7.3.2.1 Closed-loop simulation results

The control objective of CLBF-EMPC is to maximize the profit of the CSTR process of Eq. 7.26 by manipulating the inlet concentration ΔC_{A0} and the heat input rate ΔQ, while maintaining the

closed-loop state trajectories in the safe stability region $\mathscr{U}_\rho$ for all times. The objective function of the CLBF-EMPC is of the following form:

$$l_e(\tilde{x},u) = k_0 e^{-E/RT} C_A^2 \tag{7.38}$$

Additionally, a material constraint is incorporated in the CLBF-EMPC of Eq. 7.32 to make the averaged reactant material available within the entire operating period t_p to be its steady-state value, C_{A0s}. The material constraint is formulated as follows:

$$\frac{1}{t_p}\int_0^{t_p} u_1(\tau) d\tau = 0 \; kmol/m^3 \tag{7.39}$$

where the averaged reactant material in deviation form, u_1, is equal to 0. Similar to the application of CLBF-MPC scheme in Section 7.2.4, we also demonstrate the application of the RNN-based CLBF-EMPC control scheme to both bounded and unbounded unsafe regions in state-space.

We first consider the case of an unbounded unsafe region in state-space, where the unsafe region, CLBF $W_c(x)$ and the parameter values are the same as those in Section 7.2.4. The Lyapunov-based EMPC (LEMPC) that accounts for closed-loop stability only is also used here for comparison. Specifically, based on the formulation of the standard LEMPC of Eq. 3.1, the LEMPC using RNN models is presented as follows [171]:

$$\max_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(\tau),u(\tau))\, d\tau \tag{7.40a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = \frac{1}{N_e}\sum_{j=1}^{N_e} F_{nn}^j(\tilde{x}(t),u(t)) \tag{7.40b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{7.40c}$$

$$u(t) \in U, \; \forall\, t \in [t_k, t_{k+N}) \tag{7.40d}$$

$$V(\tilde{x}(t)) \leq \rho_e, \; \forall\, t \in [t_k, t_{k+N}), \text{ if } x(t_k) \in \Omega_{\rho_e} \tag{7.40e}$$

$$\dot{V}(x(t_k),u(t_k)) \leq \dot{V}(x(t_k),\Phi_{nn}(x(t_k))), \;\; \text{if } x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_e} \tag{7.40f}$$

where the notations follow those in Eq. 3.1 and Eq. 7.32. As the objective of EMPC is to dynamically optimize the profit of the CSTR process of Eq. 7.26 by maximizing the production rate $r = k_0 e^{-E/RT} C_A^2$, it is observed in Fig. 7.9 that under the LEMPC of Eq. 7.40 that does not account for safety concerns, the closed-loop state is driven to the top of the operating region where temperature is much higher than the steady-state value, to obtain an increased economic profits compared to the steady-state operation (i.e., the system is operated at steady-state for all times). Additionally, by designing the unbounded unsafe region with the form of $F(x) = x_1 + x_2$, it is noted that the temperature in the reactor plays a more important role in characterizing the unsafe region $\mathscr{D}_u$ than the concentration due to its larger magnitude. This is consistent with the operation of an exothermic reaction in CSTR, where rapid increases in temperature might lead to potential safety problems. However, it should be mentioned that reactant concentration is still accounted for in the characterization of the unbounded unsafe region $\mathscr{D}_u$ due to its impact on the reaction rate.

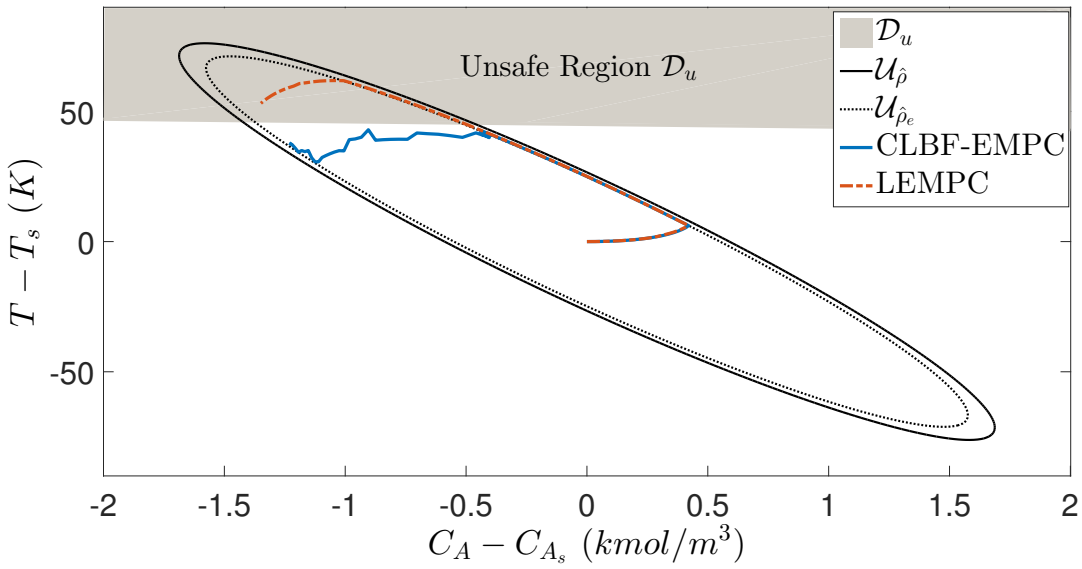


Figure 7.9: Closed-loop state trajectories for the system of Eq. 7.26 within one operating period under CLBF-EMPC and LEMPC, respectively, where the initial condition is (0, 0) and the unbounded set of unsafe states $\mathscr{D}_u$ is the gray area on the top of $\mathscr{U}_\rho$.

The closed-loop simulation results for the system of Eq. 7.26 under the RNN-based CLBF-EMPC of Eq. 7.32, and the LEMPC of Eq. 7.40 are shown in Figs. 7.9-7.11. Specifically, Fig. 7.9 shows the comparison of the state trajectories under LEMPC and CLBF-EMPC,

respectively. It is demonstrated that starting from the initial condition (0, 0), the state trajectory for one simulation period $t_p = 0.128\ hr$ under CLBF-EMPC is maintained below the unbounded unsafe region $\mathscr{D}_u$ for all times, while the one under LEMPC exceeds the threshold and enters $\mathscr{D}_u$ near the end of simulation. Additionally, we run the closed-loop simulation for successive four operating period, where each operating period is $t_p = 0.128\ hr$. The material constraint is imposed in each operating period such that the averaged reactant material (in deviation form) within each operating period equals zero. It is demonstrated in Fig. 7.10 that the state trajectory under the CLBF-EMPC of Eq. 7.32 remains in the safe stability region $\mathscr{U}_\rho$ within four operating periods, while the one under LEMPC enters the unsafe region during the first operating period and stays there for the remainder of the process operation. Both state trajectories progress in a circular manner in the stability region (the solid ellipse) because the material constraint forces the decrease of the reactant concentration near the end of each operating period. This can also be observed in the input profiles for the closed-loop system of Eq. 7.26 within four operating periods shown in Fig. 7.11, where CLBF-EMPC consumes the maximum allowable ΔC_{A0} at the beginning of each operating period and lowers the consumption near the end.

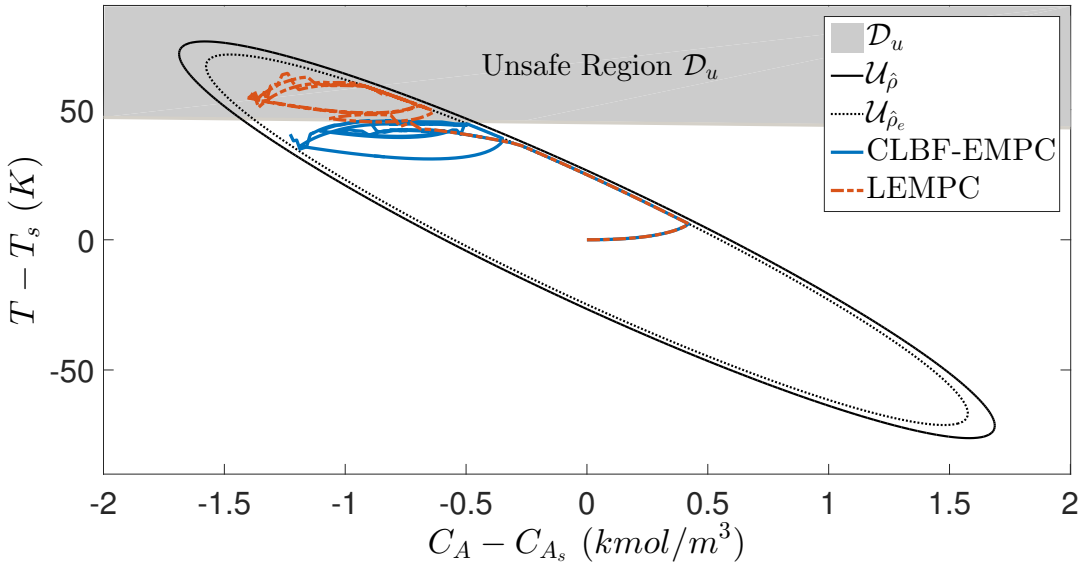


Figure 7.10: Closed-loop state trajectories for the system of Eq. 7.26 within four operating periods under CLBF-EMPC and LEMPC, respectively, where the initial condition is (0, 0) and the unbounded set of unsafe states $\mathscr{D}_u$ is the gray area on the top of $\mathscr{U}_\rho$.

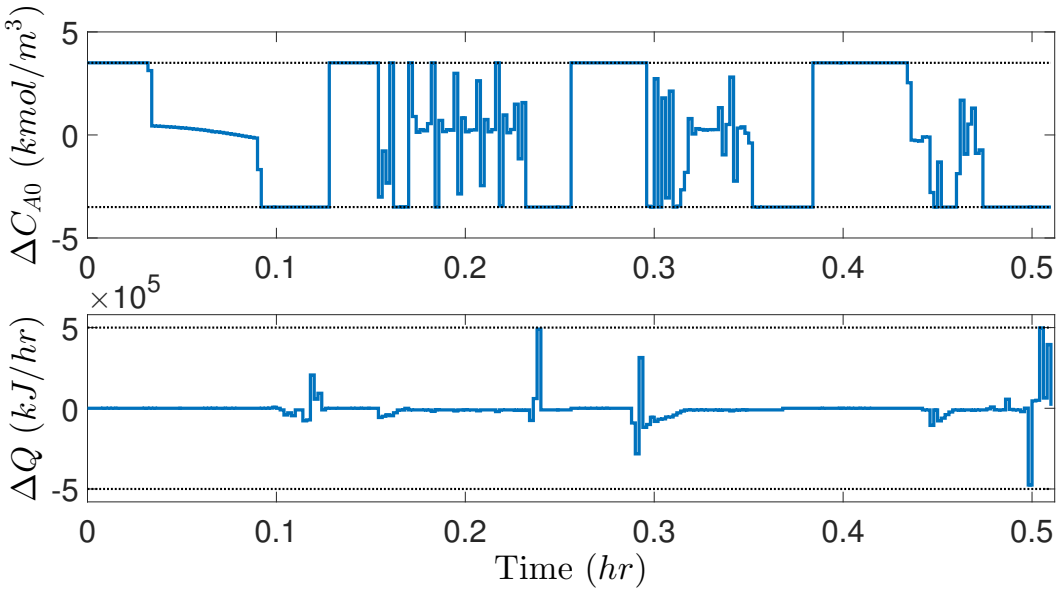


Figure 7.11: Input profiles for the closed-loop system of Eq. 7.26 within four operating periods under CLBF-EMPC, where the unsafe region is the gray area on the top of $\mathscr{U}_\rho$.

We also calculate the total economic profits over four operating periods, i.e., $L_E = \int_{t=0}^{4t_p} k_0 e^{-E/RT} C_A^2 dt$, for the closed-loop system of Eq. 7.26 under the different controllers. It was obtained that the L_E values are 8.42, 8.01 and 5.24 for the closed-loop CSTR under LEMPC, CLBF-EMPC, and steady-state operation, respectively, from which it is demonstrated that economic profits are significantly improved (around 52%) under EMPC compared to the steady-state operation. The reason for a slightly larger L_E under LEMPC than CLBF-EMPC is that the state under LEMPC enters the unsafe region during the simulation where increased production rate is obtained due to higher temperature (Fig. 7.10).

The second example is to demonstrate the effectiveness of the CLBF-EMPC of Eq. 7.32 for the CSTR system with a bounded unsafe region $\mathscr{D}_b$ in state-space. The bounded unsafe region $\mathscr{D}_b$ is designed to be a set embedded within the stability region as shown in the above example to demonstrate that the the CLBF-EMPC of Eq. 7.32 is able to achieve economic optimality while maintaining the state out of $\mathscr{D}_b$ for all times. The bounded unsafe region as well as the CLBF and its parameters are the same as those in Section 7.2.4.

The simulation results for the closed-loop system of Eq. 7.26 under CLBF-EMPC are shown

in Figs. 7.12-7.13. Specifically, in Fig. 7.12, it is demonstrated that the state trajectory under CLBF-EMPC is maintained in the safe stability region $\mathscr{U}_\rho$ for all times (i.e., four successive operating periods with $t_p = 0.128\ hr$). However, the state trajectory under LEMPC enters the bounded unsafe region $\mathscr{D}_b$ since the design of the LEMPC of Eq. 7.40 does not account for any safety constraints. Similarly, Fig. 7.13 shows the input profiles for the closed-loop system of Eq. 7.26 within four operating periods under the CLBF-EMPC of Eq. 7.32, where ΔC_{A0} shows variation due to the material constraint of Eq. 7.39 applied in each operating period. Additionally, the accumulated economic profits are calculated for the closed-loop system of Eq. 7.26 in the presence of a bounded unsafe region. It was found that the L_E values are 8.42, 8.47 and 5.24 for LEMPC, CLBF-EMPC, and steady-state operation, respectively. This again demonstrates that process economics is optimized under EMPC while closed-loop stability and process operational safety are both guaranteed. It is noted that the total economic profits under LEMPC and under CLBF-EMPC are very close since the two state trajectories both stay in a region above the unsafe set for most of the simulation time (Fig. 7.12). The only difference is that the state trajectory under CLBF-EMPC avoids the bounded unsafe region for all times, while the one under LEMPC does not.

Additionally, it is noted that the RNN-based MPC is computationally more demanding than the first-principles-model-based MPC because the RNN model is essentially a complicated nonlinear function which requires more computation time for prediction. In our example, the computation time for running RNN-based MPC is around 2.3 s, which is less than one sampling period (i.e., 2×10^{-3} hr = 7.2 s) such that it can implemented in real-time optimization and control. The above case studies demonstrate that the CLBF-EMPC of Eq. 7.32 based on an ensemble of RNN models achieved desired model prediction results for the nonlinear system of Eq. 7.26, and thus, is able to optimize control actions that maintain the closed-loop state within the safe stability region $\mathscr{U}_\rho$ for all times. Additionally, we demonstrate the applicability of the CLBF-EMPC of Eq. 7.32 to both bounded and unbounded unsafe regions in a CSTR example. The economic profits over multiple operating periods are calculated and compared under LEMPC, CLBF-EMPC and

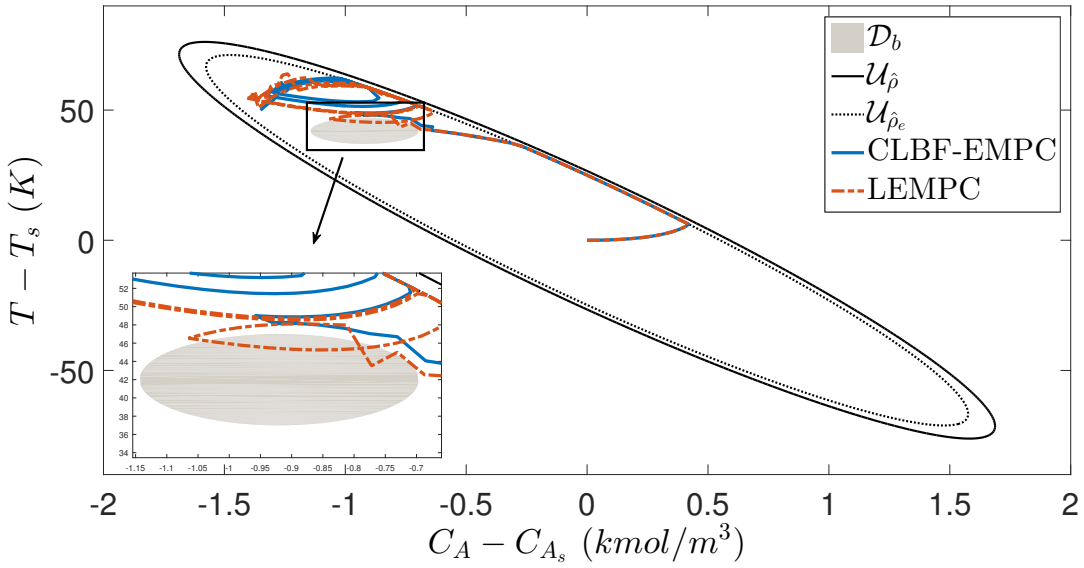


Figure 7.12: Closed-loop state trajectories for the system of Eq. 7.26 within four operating periods under CLBF-EMPC and LEMPC, respectively, where the initial condition is (0, 0) and the bounded set of unsafe states $\mathscr{D}_b$ is embedded within $\mathscr{U}_\rho$.

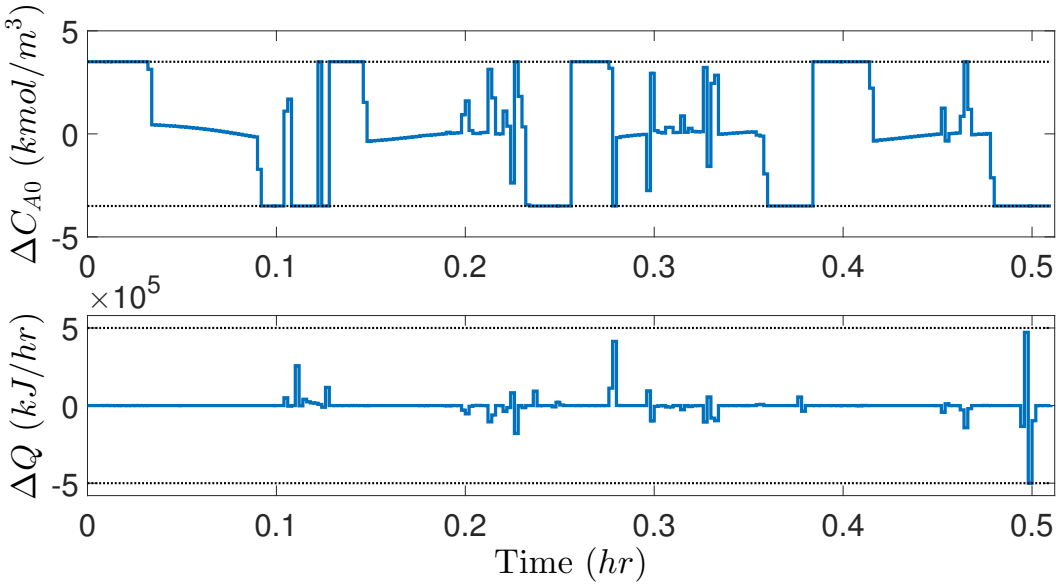


Figure 7.13: Input profiles for the closed-loop system of Eq. 7.26 within four operating periods under CLBF-EMPC, where the bounded set of unsafe states $\mathscr{D}_b$ is embedded within $\mathscr{U}_\rho$.

steady-state operation, respectively, from which it can be concluded that significant improvement of economic benefits can be achieved under EMPC.

7.3.2.2 Real-time CLBF-EMPC with online learning of RNN models

The closed-loop simulation results for the CSTR of Eq. 7.26 under the machine-learning-based CLBF-EMPC of Eq. 7.32 with and without online learning of RNN models, respectively, are shown in this subsection. The disturbance on the feed flow rate F which varies from 5 m^3/h to 10 m^3/h at $t = 0.11$ hr is introduced into the closed-loop system. The simulation results are shown in Figs. 7.14-7.17. In Fig. 7.14, it is demonstrated that the closed-loop state trajectory under CLBF-EMPC with updating RNN models avoids the unsafe region while the one under the CLBF-EMPC using the initial RNN model for all times enters the unsafe region $\mathscr{D}$ near the end of operating period due to the disturbed feed flow rate F and reaction rate.

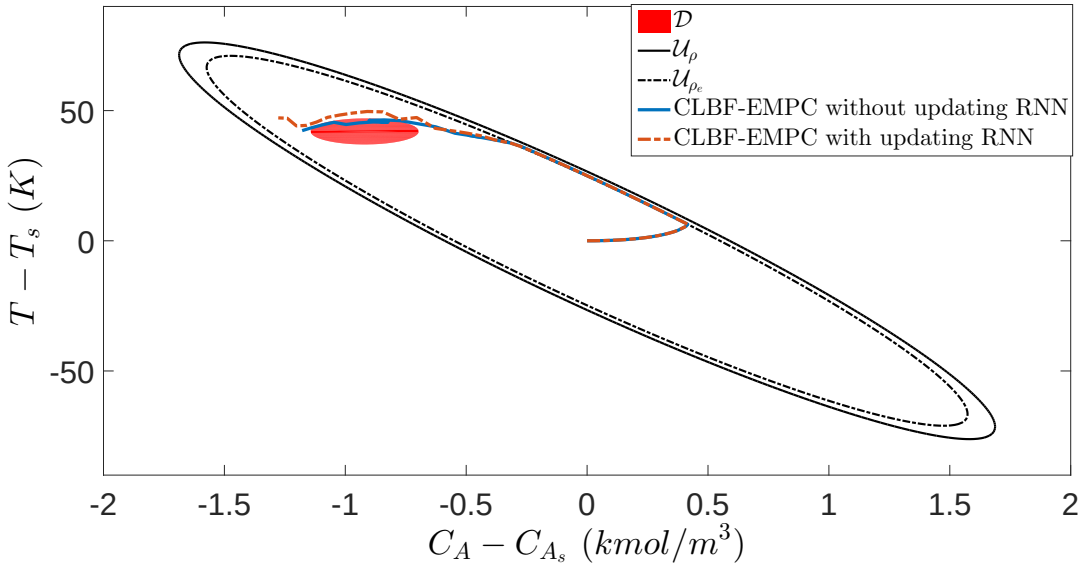


Figure 7.14: The state-space profiles for the closed-loop CSTR subject to time-varying disturbances under CLBF-EMPC with (red trajectory) and without online RNN update (blue trajectory), respectively, for an initial condition (0,0).

Moreover, the closed-loop simulation of the CSTR system under CLBF-EMPC with multiple operating periods is performed with the following disturbances: (1) the feed flow rate F is changing from 5 m^3/h to 11.5 m^3/h at $t = 0.1$ hr during the first operating period from $t = 0$ hr to $t = 0.128$ hr, and (2) the actual value of the pre-exponential constant k_0 used in the process model is reduced by 20% to represent a change in the reaction rate at $t = 0.148$ hr during the

second operating period from $t = 0.128\ hr$ to $t = 0.256\ hr$. Fig. 7.15 and Fig. 7.16 show the closed-loop simulation results under the above settings. Specifically, Fig. 7.15 demonstrates that with online learning of RNN models, the closed-loop state trajectory under CLBF-EMPC is able to avoid the unsafe region for all times within two consecutive EMPC operating periods. Fig. 7.16 shows the corresponding input profiles under CLBF-EMPC, from which it is observed that the inlet concentration ΔC_{A0} consumes its maximum allowable value at the beginning of each operating period, and thus, decreases to its lower bound near the end of each operating period to meet the material constraint of Eq. 7.39. Additionally, the accumulated prediction error diagram under CLBF-EMPC with and without online learning of RNN models is shown in Fig. 7.17. It is demonstrated that the prediction error (red lines) for the CLBF-EMPC with updating RNN models is maintained at a very low level during the two consecutive EMPC operating periods. However, the prediction error (blue lines) derived from the CLBF-EMPC without updating RNN models indicates a large model mismatch between the initial RNN model for the nominal CSTR of Eq. 7.26 and the actual disturbed system.

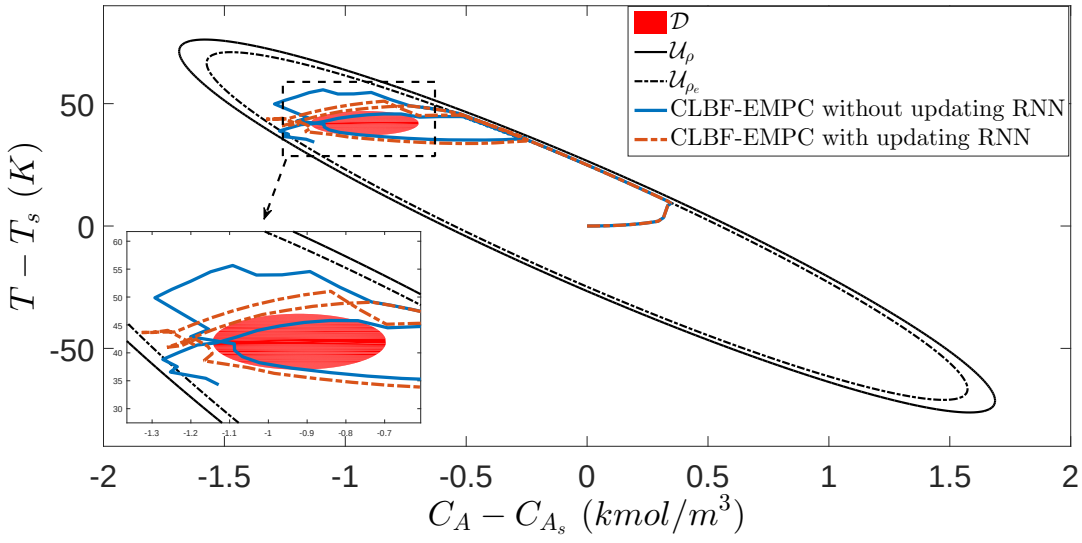


Figure 7.15: The state-space profiles for the closed-loop CSTR subject to time-varying disturbances under CLBF-EMPC with (red trajectory) and without online RNN update (blue trajectory), respectively, for two consecutive operating periods with an initial condition (0,0).

Lastly, to demonstrate the improved process economic benefits under the time-varying

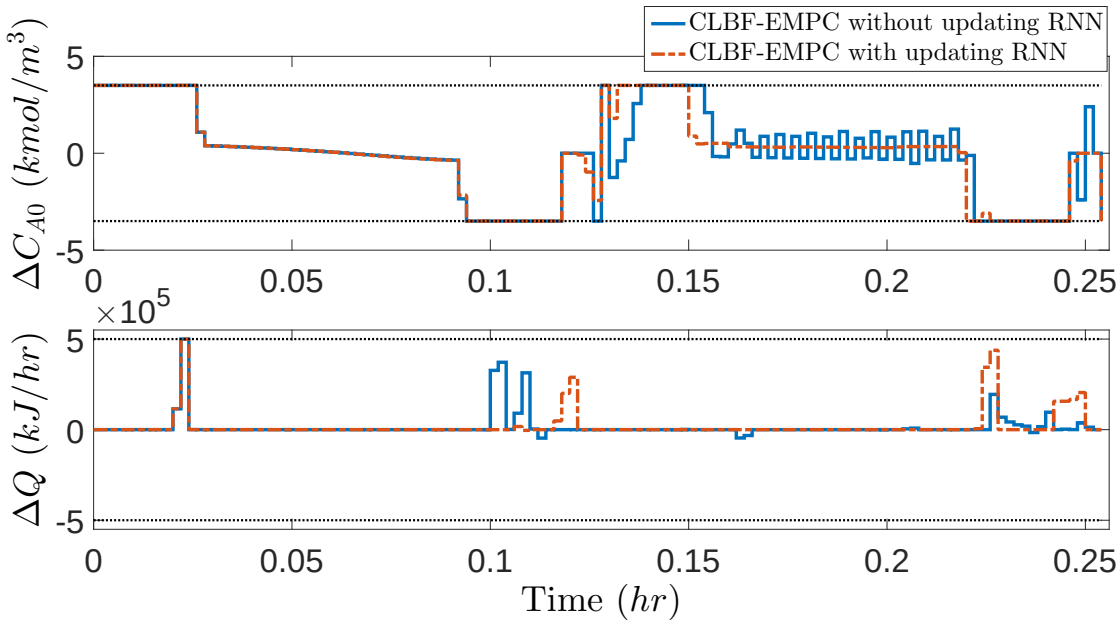


Figure 7.16: Manipulated input profiles ($u_1 = \Delta C_{A0}$, $u_2 = \Delta Q$) for the closed-loop CSTR subject to time-varying disturbances under CLBF-EMPC with (red trajectory) and without online RNN update (blue trajectory), respectively, for two consecutive operating periods with an initial condition (0,0).

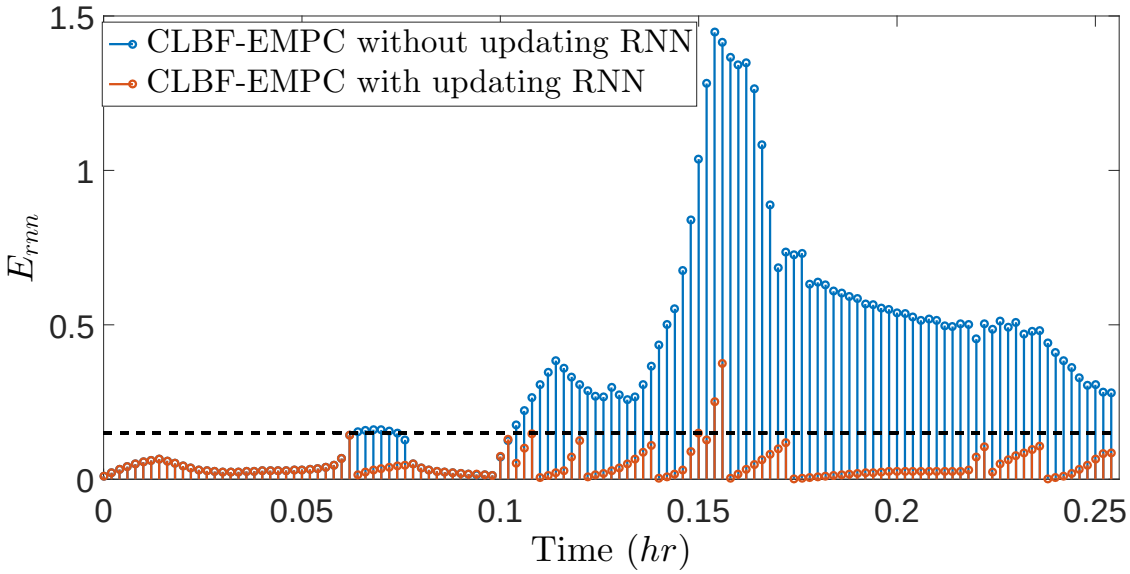


Figure 7.17: Value of $E_{rnn}(t)$ at each sampling time for the closed-loop CSTR subject to time-varying disturbances under CLBF-EMPC with and without online RNN update, respectively, where the threshold E_T is set to 0.15.

operation of EMPC, accumulated economic profits over the entire operating period, i.e., $L_E = \int_0^{t=0.256\ hr} l_e(x,u)d\tau$ is compared for the CLBF-EMPC and the steady-state operation (i.e., the CSTR of Eq. 7.26 is operated at the steady-state for all times). It is obtained that $L_E = 4.93$ for the closed-loop system under CLBF-EMPC with online update of RNN models and $L_E = 2.61$ for the steady-state operation within $0.256\ hr$. Therefore, it is concluded that closed-loop stability, process operational safety and economic optimality are achieved simultaneously for the disturbed CSTR process of Eq. 7.26 under the CLBF-EMPC of Eq. 7.32 with online learning of RNN models.

7.4 Conclusions

In this chapter, CLBF-MPC and CLBF-EMPC methods formulated with machine learning models were developed for nonlinear process systems. RNN models were first trained using extensive open-loop simulation data to capture process dynamics in a certain operating region such that the modeling error between the recurrent neural network model and the actual nonlinear process model was sufficiently small. Then, the well-fitting RNN models were incorporated in the formulation of CLBF-MPC/EMPC to predict process dynamics, for which ensemble learning was employed to improve prediction accuracy and parallel computing was used to reduce computation time of multiple RNN models.

The stability analysis of the closed-loop system under the CLBF-MPC/EMPC schemes using RNN models established the boundedness of the closed-loop state in the safety and stability region for MPC and EMPC and demonstrated the ultimate convergence to a small neighborhood around the origin for MPC. Additionally, event-triggered and error-triggered mechanisms were designed for the real-time implementation of CLBF-MPC and CLBF-EMPC schemes to update the RNN models online using the most recent process data that account for nonlinear dynamics in the presence of time-varying disturbances. The application of the machine-leaning-based control schemes to a chemical reactor demonstrated the applicability and effectiveness of the schemes in stabilizing nonlinear systems with simultaneous stability and safety guarantees, and the ability

to deal with time-varying disturbances using online learning of machine learning models.

Chapter 8

Detector-Integrated Controller for Process Cybersecurity

8.1 Introduction

Cyber-physical systems (CPS) integrate communication networks, computation, and physical process components to ensure automated real-time operation in a seamless manner. Stable and secure operation of cyber-physical systems require accurate information and reliable communication technologies. In more recent years, the cyber-security of cyber-physical systems has become increasingly important as more communication networks are replaced or complemented by wireless networks in addition to point-to-point communications [3, 35]. While these new developments increase operation efficiency and performance, they also increase the system's vulnerability to cyber-attacks. As more components are included, there is a high probability that continuous feedback measurements cannot be guaranteed due to bursts of network transmission errors, which poses a challenge for closed-loop control systems that rely on accurate feedback measurements. Malicious cyber-attacks could target any device or communication channels in the control network to modify control actions and jeopardize operational cost, stability, integrity, and other safety considerations. With access to technical details of the control system,

these targeted cyber-attacks are intelligently designed to disrupt process operation and compromise fundamental process safety. As cyber-attacks pose severe threats to the control system, safety measures addressing cyber-security need to be carefully considered and incorporated in plant-wide risk assessments.

On the other hand, with the increase in digital connectivity and computing power, potential applications of archived plant data could extend beyond day-to-day monitoring and operation. One example use of these "big data" approaches is cyber-attack and anomaly detection. Due to the close interactions between cyber and physical components, operational cyber-security of control systems would mandate a different strategy than traditional information technology (IT) approaches – one that combines robust control strategies with an advanced detection scheme using the process data at hand [128]. Due to the sophistication of cyber-attacks and their accessibility to control system information, they are intended to disrupt the closed-loop system while avoiding being detected by conventional detection methods or by control engineers, thus making them fundamentally different from sensor or actuator faults. Situations where conventional model-based detection schemes may be rendered ineffective by intelligent cyber-attacks can be potentially tackled by data-based detection methods [27]. Machine learning, a method of data analysis that can help engineers learn from data, identify patterns and make decisions with minimal human intervention, has attracted an increasing attention and shown promising potential for use in detection of cyber-attacks. The development and applications of machine-learning methods in traditional engineering fields have increased in recent years, and more specifically in the field of systems engineering, e.g., [119, 130, 155, 181]. Machine learning techniques, such as artificial neural networks, support vector machines, as well as more advanced deep learning methods, such as recurrent neural networks, have demonstrated success in detecting machine and plant anomalies, e.g., [24, 28, 29, 59, 105, 114, 149, 163, 168], and can be readily adopted in the context of control theory and cyber-physical security.

In this chapter, machine-learning-based detection systems and resilient control schemes are developed to detect and mitigate the impact of stealthy cyber-attacks. In the first section, the

concept of stealthy cyber-attacks is presented, followed by several common cyber-attacks in literature. The second section presents the construction of data-based machine-learning detection algorithms which can effectively detect multiple classes of intelligent cyber-attacks. Subsequently, we design several resilient control strategies to promptly contain and eliminate the impact of cyber-attacks upon detection. The application to a benchmark multivariable nonlinear process example is presented to evaluate the ability of the proposed detection and mitigation schemes.

8.1.1 Notation

The set of real numbers is denoted by $\mathbf{R}$, and the set of nonnegative real numbers is denoted by $\mathbf{R}_+$. $\mathbf{R}^n$ is an n-dimensional real (Euclidean) space. The notation $|\cdot|$ is used to denote the Euclidean norm of a vector, and the notation $|\cdot|_Q$ denotes a weighted Euclidean norm of a vector (i.e., $|x|_Q = \sqrt{x^T Q x}$ where Q is a positive definite matrix). x^T denotes the transpose of x. The notation $L_f V(x)$ denotes the standard Lie derivative of function $V(x)$ with respect to the vector field f, i.e., $L_f V(x) := \frac{\partial V(x)}{\partial x} f$. A scalar continuous function $V : \mathbf{R}^n \to \mathbf{R}$ is proper if the set $\{x \in \mathbf{R}^n \mid V(x) \leq k\}$ is compact for all $k \in \mathbf{R}$, or equivalently, V is radially unbounded in the sense that $\lim_{|x|\to+\infty} V(x) = +\infty$ holds.

For given positive real numbers β and ε, $\mathscr{B}_\beta(\varepsilon) := \{x \in \mathbf{R}^n \mid |x-\varepsilon| < \beta\}$ is an open ball around ε with radius of β. The relative complement of the set A in B is denoted by $A \backslash B := \{x \in A, x \notin B\}$. A function $f(\cdot)$ is of class $\mathscr{C}^1$ if it is continuously differentiable. Given a set $\mathscr{D}$, the boundary, the closure, and the interior of $\mathscr{D}$ are denoted by $\partial\mathscr{D}$, $\overline{\mathscr{D}}$, and Int($\mathscr{D}$), respectively. A continuous function $\alpha : [0, a) \to \mathbf{R}_+$ is said to be of class $\mathscr{K}$ if it is strictly increasing and $\alpha(0) = 0$.

8.1.2 Class of Nonlinear Systems

The class of continuous-time nonlinear systems considered is described by the following state-space form:

$$\dot{x}(t) = f(x(t), u(t), w(t)) \tag{8.1a}$$

$$\bar{x}(t) = h(x(t)) \tag{8.1b}$$

where $x(t) \in D \subset \mathbf{R}^n$ is the state vector, and $u(t) \in \mathbf{R}^m$ is the manipulated input vector, which is constrained by $u \in U := \{u_i^{min} \leq u_i \leq u_i^{max}, i = 1, ..., m\} \subset \mathbf{R}^m$, where u_i^{min} and u_i^{max} are the lower and upper bounds for the input vector. We will denote the vector of state measurements from sensors, which may be compromised by sensor cyber-attacks, with $\bar{x}(t) \in \mathbf{R}^n$. When no cyber-attacks are present in the system, $\bar{x}(t) = x(t)$. $w \in W$ is the noise vector, where $W := \{w \in \mathbf{R}^l : \ | |w| \leq \theta, \ \theta \geq 0\}$. Without loss of generality, the initial time t_0 is taken to be zero ($t_0 = 0$). It is assumed that $f(\cdot, \cdot, \cdot)$ is a sufficiently smooth vector function of its arguments, and $h(\cdot)$ is a sufficiently smooth vector function of x where $f(0,0,0) = 0$, $h(0) = 0$. Thus, the origin is an equilibrium point of the system of Eq. 8.1 under $u(t) = 0$.

We assume that there exists an explicit feedback controller of the form $u = \Phi(x) \in U$ that can render the origin of the nominal closed-loop system of Eq. 8.1 (i.e., $w(t) \equiv 0$) asymptotically stable. The stabilizability assumption implies the existence of a $\mathscr{C}^1$ Lyapunov function $V : D \to \mathbf{R}_+$ that satisfies the following conditions:

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|), \tag{8.2a}$$

$$\frac{\partial V(x)}{\partial x} f(x, \Phi(x), 0) \leq -\alpha_3(|x|), \tag{8.2b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq \alpha_4(|x|) \tag{8.2c}$$

where D is an open neighborhood around the origin, and $\alpha_i(\cdot)$, $i = 1, 2, 3, 4$, are class $\mathscr{K}$ functions.

The stability region Ω_ρ of the closed-loop system of Eq. 8.1 is characterized as a level set of $V(x)$ inside D in which Eq. 8.2 is satisfied under $u = \Phi(x) \in U$, i.e., $\Omega_\rho := \{x \in D \mid V(x) \leq \rho, \rho > 0\}$. Therefore, given that the sensor measurements received by the controller are secure and reliable (i.e., $\bar{x}(t) = x(t)$), the controller $u = \Phi(x) \in U$ guarantees that the state trajectory of the closed-loop system of Eq. 8.1 remains within Ω_ρ and asymptotically converges to the origin for any initial conditions $x_0 \in \Omega_\rho$. Additionally, the smoothness property of $f(x,u,w)$ and the boundedness of $u \in U$ imply that there exist positive constants M, L_x, $L_x^{'}$ such that the following inequalities hold for all x, x' in a neighborhood around the origin:

$$|f(x,u,0)| \leq M \tag{8.3a}$$

$$|f(x,u,0) - f(x',u,0)| \leq L_x|x - x'| \tag{8.3b}$$

$$\left|\frac{\partial V(x)}{\partial x} f(x,u,0) - \frac{\partial V(x')}{\partial x} f(x',u,0)\right| \leq L_x^{'}|x - x'| \tag{8.3c}$$

8.1.3 Lyapunov-based MPC and EMPC

Cyber-attack detection systems and resilient control schemes in this chapter are developed in the context of model predictive control, and more specifically, Lyapunov-based model predictive control (LMPC) and Lyapunov-based economic model predictive control (LEMPC). Therefore, the LMPC and LEMPC formulations are presented here again for convenience. Specifically, the

LMPC optimization problem is formulated as follows:

$$\min_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} l_t(\tilde{x}(\tau),u(\tau))\,d\tau \tag{8.4a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t),u(t),0) \tag{8.4b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{8.4c}$$

$$u(t) \in U,\ \forall\, t \in [t_k,t_{k+N}) \tag{8.4d}$$

$$\frac{\partial V(x(t_k))}{\partial x} f(x(t_k),u(t_k),0) \leq \frac{\partial V(x(t_k))}{\partial x} f(x(t_k),\Phi(x(t_k)),0),$$

$$\text{if } x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{min}} \tag{8.4e}$$

$$V(\tilde{x}(t)) \leq \rho_{min},\ \forall\, t \in [t_k,t_{k+N}),\ \text{if } x(t_k) \in \Omega_{\rho_{min}} \tag{8.4f}$$

where the notations follow those in Eq. 2.19. The LEMPC is represented by the following optimization problem:

$$\max_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(\tau),u(\tau))\,d\tau \tag{8.5a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t),u(t),0) \tag{8.5b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{8.5c}$$

$$u(t) \in U,\ \forall\, t \in [t_k,t_{k+N}) \tag{8.5d}$$

$$V(\tilde{x}(t)) \leq \rho_e,\ \forall\, t \in [t_k,t_{k+N}),\ \text{if } x(t_k) \in \Omega_{\rho_e} \tag{8.5e}$$

$$\frac{\partial V(x(t_k))}{\partial x} f(x(t_k),u(t_k),0) \leq \frac{\partial V(x(t_k))}{\partial x} f(x(t_k),\Phi(x(t_k)),0),$$

$$\text{if } x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_e} \tag{8.5f}$$

where the notations follow those in Eq. 3.1. For EMPC, it is common that chemical processes are subject to periodic feed stock constraints, which are specified as part of the input constraint set U, where the quantity of feed materials is limited within a fixed period of time t_{N_p}. During this period

of time, the total feed material is constrained to a constant value C as follows:

$$\frac{1}{t_{N_p}} \int_{t_0}^{t_{N_p}} u_m(\tau) d\tau = C \tag{8.6}$$

where u_m represents feed material used at every sampling period. Therefore, the material consumption constraint renews every t_{N_p}. If the total operation time is longer than one material constraint period, this material consumption constraint results in cyclic operation of the plant, and consequently, cyclic behavior of the state-space trajectory. At the start of a new material constraint period, the total consumption limit is renewed, as new feed materials become available to be used again for the next constraint period.

When a secure state measurement x is available every sampling step, it is demonstrated that closed-loop stability is guaranteed for the nonlinear system of Eq. 8.1 in the sense that for any initial condition $x_0 \in \Omega_\rho$, the closed-loop state is guaranteed to be bounded in Ω_ρ for all times under LMPC/LEMPC, and can be ultimately driven to a small neighborhood $\Omega_{\rho_{min}}$ around the origin under LMPC. However, under cyber-attacks that compromise sensor measurements or communication networks between sensors and controllers, closed-loop stability under LMPC/LEMPC is no longer guaranteed because the evolution of the true state will be different from the MPC predicted state trajectory based on falsified state measurements.

8.2 Intelligent Cyber-Attacks

Stealthy, intelligent cyber-attack diagnosis and defense span a much broader scope than classical fault detection problems because intelligent adversaries can modify the actuator, the sensor, or the control implementation using process and control system information. With knowledge of the plant model and of the control formulation, cyber-attacks are strategically programmed with the goal of disruption, and are fundamentally different from ordinary sensor and actuator faults. Specifically, among sensor cyber-attacks, Denial-of-Service attacks, replay attacks and deception attacks (e.g., Min-Max, Geometric, Surge) are some of the most common and easily implementable

ones by attackers [140]. They are designed to intentionally destroy the control objectives of the system, disrupting system stability and degrading control performance. Furthermore, the effects of these attacks may be only observed in changes of the dynamic behavior (runtime variables) of the closed-loop system; thus, using hardware performance counters to track code modifications is not feasible [75].

In this section, we consider the attacks on sensor measurements. Sensor feedback measurements must accurately report the true state of the process to ensure closed-loop stability; falsified measurements may result in control actions that will no longer achieve maximum economic benefit and may ultimately drive the true process states outside of the stability region. There are some standard types of cyber-attacks considered in literature [140]. For example, min-max cyber-attacks aim to achieve maximum disruptive impact within shortest amount of time. Surge attacks cause maximum deviation for an initial "surge" period, and then the attacked value is set to a reduced value for the remainder of the attack duration such that the cumulative deviation will not exceed a certain threshold that will trigger alarms in conventional detection methods such as Cumulative Sum [27, 100]. Geometric attacks geometrically increase the deviation of the attacked value from its true value until it reaches the alarming threshold. Being process and controller behavior aware, the cyber-attacks will have access to information on the operating region of the process under LMPC/LEMPC, and existing alarms configured on the input and output ranges. Specifically, when attacks intend to induce maximum disruption (i.e., in min-max or surge attacks), the attacked value will be set to the maximum or minimum value beyond which an alarm monitoring the current state measurement will be immediately triggered. These intelligent cyber-attacks are designed such that no alarms will be sounded (i.e., the falsified state measurement is not outside the operating stability region or the alarm window) and the controller is still able to compute feasible control actions, but have large enough variations such that economic optimality and closed-loop stability will be lost.

8.2.1 Types of Intelligent Cyber-attakcs

Consider the system of Eq. 8.1 under LMPC/LEMPC within the operating region Ω_ρ. The cyber-attacks imposed on the sensors are designed to prevent having a falsified measurement beyond the operating region Ω_ρ and to avoid triggering any immediate alarms based on the values of the state measurements. The mathematical formulations of min-max, surge, geometric, and replay attacks are presented as follows.

8.2.1.1 Min-max cyber-attack

While avoiding triggering any alarms, min-max attacks result in maximum destabilizing impact within a short time period. Therefore, the falsified state measurements take values that are furthest from the equilibrium point (minimum or maximum) but not outside of the operating region Ω_ρ. The min-max attack can be formulated as follows:

$$\bar{x}(t_i) = \min_{x\in\mathbf{R^n}} / \max_{x\in\mathbf{R^n}} \{x \mid V(x(t_i)) = \rho\}, \;\; \forall\, i \in [i_0, i_0 + L_a] \tag{8.7}$$

where ρ defines the level set of the Lyapunov function $V(x)$ that characterizes the operating region of the closed-loop system of Eq. 8.1 under LMPC/LEMPC, $\bar{x}$ is the compromised sensor measurement, i_0 is the time instant that the attack is introduced, and L_a is the total duration of the attack in terms of sampling periods.

8.2.1.2 Geometric cyber-attack

Under geometric cyber-attacks, closed-loop system stability deteriorates at a geometric speed until the cyber-attack reaches the maximum or minimum allowable value as characterized by the operating region. At the start of the attack t_{i_0}, a small constant $\beta \in \mathbf{R}$ is added to the true measured output $x(t_{i_0})$, where $x(t_{i_0}) + \beta$ is well below the alarm threshold. Following that, at each subsequent time step, β is multiplied by a factor $(1+\alpha)$, where $\alpha \in (0,1)$, until $\bar{x}$ reaches the maximum allowable attack value bounded by Ω_ρ. Thus, attackers will choose the two parameters α and β

based on Ω_ρ and the attack duration. Geometric attacks can be written in the form as follows:

$$\bar{x}(t_i) = x(t_i) + \beta \times (1+\alpha)^{i-i_0}, \ \ \forall\, i \in [i_0, i_0 + L_a] \tag{8.8}$$

where β and α are parameters that define the magnitude and speed of the geometric attack.

8.2.1.3 Replay cyber-attack

Replay cyber-attacks have access to all previous system outputs corresponding to secure nominal operating conditions where no cyber-attacks are present. The attacker extracts segments of these previous state measurements and injects them into the current measurement readings. As the replayed values are given by secure sensors and supposedly inside the operating bounds, classical detectors will not be able to recognize any abnormalities. Replay attacks can be represented by the following equations:

$$\bar{x}(t_i) = x(t_k), \ \forall\, k \in [k_0, k_0 + L_a], \ \ \forall\, i \in [i_0, i_0 + L_a] \tag{8.9}$$

where $x(t_k)$ is the true plant measurement, L_a represents the length of the attack (which is also the length of the replay segment) in terms of sampling periods, and $\bar{x}$ is the series of replay attacks added at time t_{i_0} duplicating previous state measurements that are recorded starting from time t_{k_0}. The duration of the attack could be exactly the length of one or more material constraint periods. Therefore, the tampered state trajectory would look identical to the nominal state trajectory of one (or more) complete cycle(s) of operation starting from a different set of initial conditions.

8.2.1.4 Surge cyber-attack

Surge cyber-attack is a stealthy cyber-attack that cannot be detected by conventional detection methods such as cumulative sum (CUSUM). Specifically, based on the process model of Eq. 8.1, CUSUM statistic detection method [27] is developed to minimize the detection time when a

cyber-attack occurs. The CUSUM statistic method detects cyber-attacks by calculating the cumulative sum of the deviation between expected and measured states as follows:

$$S(k) = (S(k-1) + z(k))^{+},\ S(0) = 0 \tag{8.10a}$$

$$D(S(k)) = \begin{cases} 1, & \text{if } S(k) > S_{TH} \\ 0, & \text{otherwise} \end{cases} \tag{8.10b}$$

where $S(k)$ is the nonparametric CUSUM statistic and S_{TH} is the threshold of the detection of cyber-attacks. $(S)^{+} = S$, if $S \geq 0$ and $(S)^{+} = 0$ otherwise. D is the detection indicator where $D = 1$ indicates that the cyber-attack is confirmed or there is no cyber-attack if $D = 0$. $z(k)$ is the deviation between expected states $\tilde{x}(t_k)$ and measured states $x(t_k)$ at time $t = t_k$: $z(k) := |\tilde{x}(t_k) - x(t_k)| - b$ where $\tilde{x}(t_k)$ is derived using the known process model, the state and the control action at $t = t_{k-1}$, and b is a small positive constant to reduce the false alarm rate due to disturbances.

With a carefully selected S_{TH}, the model-based detection method can detect many sensor cyber-attacks efficiently. However, the above model-based method may be evaded and becomes invalid for stealthy cyber-attacks if attackers know more about the system (e.g., the system model and the principles of the detection method). For example, surge attacks maximize the disruptive impact for an initial short period of time, then they remain at a lower value for the rest of the attack duration to maintain $S(k)$ below S_{TH}. The maximum or minimum attack value is also defined based on the operating region, Ω_{ρ}. The length of the initial surge period and the reduced value after the surge can be designed in many ways as long as the cumulative error from t_{i_0} to $t_{i_0+L_a}$ between state measurements and their predicted true values does not exceed the threshold S_{TH} of the CUSUM detection method. In this study, the reduced value after the surge is set to act as a sufficiently small bounded noise imposed on the attacked sensor. The formulation of the surge attack is presented below:

$$\begin{aligned} \bar{x}(t_i) &= \min_{x \in \mathbf{R^n}} / \max_{x \in \mathbf{R^n}} \{x \mid V(x(t_i)) = \rho\},\ \text{if } i_0 \leq i \leq i_0 + L_s \\ \bar{x}(t_i) &= x(t_i) + \eta(t_i),\ \text{if } i_0 + L_s < i \leq i_0 + L_a \end{aligned} \tag{8.11}$$

where i_0 is the start time of the attack, L_s is the duration of the initial surge, and $\eta_l \leq \eta(t_u) \leq \eta_u$ is the bounded noise added on the sensor measurement after the initial surge period, where η_l and η_u are the lower and upper bounds of the noise, respectively.

8.3 Detection of Cyber-Attacks Targeting MPC Systems

While conventional detection methods have demonstrated their effectiveness in detecting suspicious process variable deviations, most of these methods are model-based – either dependent on network and computer system models, or on physical process models. Certain classes of intelligent cyber-attacks either render traditional detection methods ineffective, or remain undetected until the system experiences a significant deviation and reaches an undesirable operating point, at which the existing alarm systems could be triggered. The goal of a robust cyber-attack detector is to identify attacks from subtle variations in real-time process state measurements and mitigate the risk before an operation alarm is triggered. Therefore, without explicit knowledge on the process model, adopting a data-based detection approach utilizing machine-learning algorithms provides a promising path for the detection of unknown intelligent cyber-attacks. The integration of existing advanced control techniques (e.g., MPC) and online machine-learning-based detection algorithms adds another protective safeguard to the multi-layer cyber-defense strategy that is standard to next-generation smart manufacturing. Cyber-attack detection carried out using machine-learning methods have been studied in many literature [2, 62, 112]. Using data-based methods to train a detection algorithm for cyber-attacks separates the detector from the physical process model, and therefore makes the detector resilient to both process changes and intelligent stealthy attacks designed based on process behavior. Amongst advanced machine-learning methods, neural networks (NN) have been successful in a wide range of applications for both supervised and unsupervised classifications [53]. In a supervised classification problem, by training the neural network with labeled data corresponding to each target class, the neural network can be used to classify new data into classes that share similar

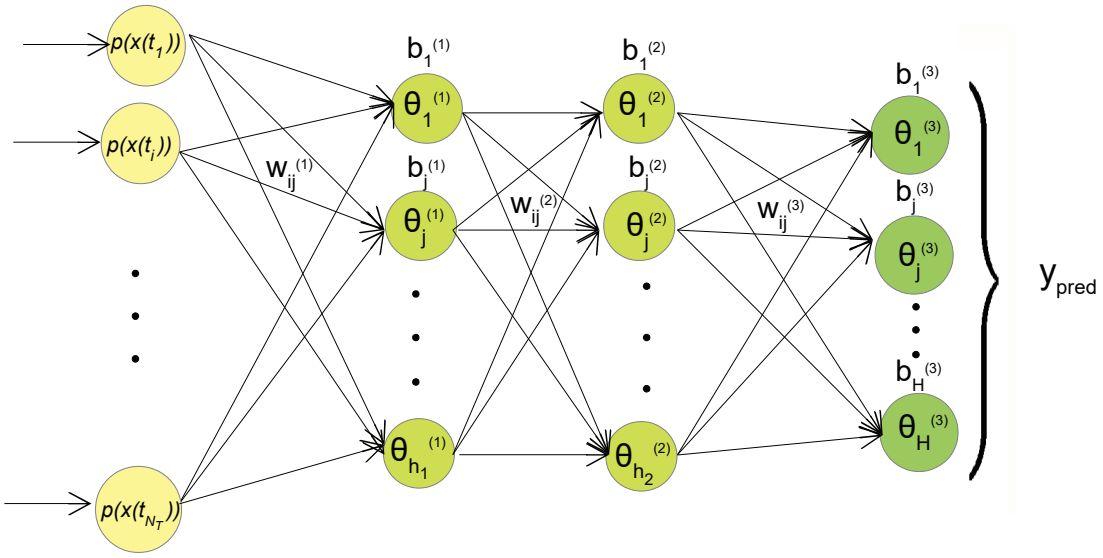


Figure 8.1: Feed-forward neural network structure with 2 hidden layers with inputs being a nonlinear function $p(\bar{x})$ at each sampling time of the model predictive controller within the detection window N_T, and output being the probability of each class label for the examined trajectory indicating the status and/or type of cyber-attack.

characteristics. Depending on the training data, the neural network can distinguish between two (i.e., "attack" or "no attack") or multiple classes (each class representing a known type of attack).

We use a feed-forward artificial neural network for supervised classification in this study. Each layer in the neural network consists of a series of nonlinear functions, yielding values for the neurons in the subsequent layer from the previous layer. Specifically, the neurons in the first hidden layer are derived from the inputs, and the neurons in the output layer are calculated from those in the last hidden layer. These nonlinear functions are activation functions of the weighted sum of inputs (or neurons in the previous layer) with an added bias term. The structure of a basic neural network model employed here is shown in Fig. 8.1, with each input representing a nonlinear function $p(\cdot)$ of the full state measurements at each sampling time, and an output vector for predicted class label. The mathematical formulation of a two-hidden-layer feed-forward neural

network is as follows:

$$\theta_j^{(1)} = g_1(\sum_{i=1}^{N_T} w_{ij}^{(1)} p(\bar{x}(t_i)) + b_j^{(1)}) \tag{8.12a}$$

$$\theta_j^{(2)} = g_2(\sum_{i=1}^{h_1} w_{ij}^{(2)} \theta_i^{(1)} + b_j^{(2)}) \tag{8.12b}$$

$$\theta_j^{(3)} = g_3(\sum_{i=1}^{h_2} w_{ij}^{(3)} \theta_i^{(2)} + b_j^{(3)}), \;\; y_{pred} = [\theta_1^{(3)}, \theta_2^{(3)}, ..., \theta_H^{(3)}]^T \tag{8.12c}$$

with $\theta_j^{(1)}$ and $\theta_j^{(2)}$ representing neurons in the first and second hidden layer, respectively, where $j = 1, ..., h_l$ is the number of neurons in layer $l = 1$ and $l = 2$. $\theta_j^{(3)}$ represents neurons in the output layer ($l = 3$), where $j = 1, ..., H$, and H is the number of class labels. In this study, we use two hidden layers for the cyber-attack detector design; however, multiple hidden layers can also be developed using similar formulations. For each sample, the input layer consists of variables $p(\bar{x}(t_i))$, which is a nonlinear function of the full-state measurements at time t_i, where $i = 1, ..., N_T$ is the length of the time-varying trajectory. The weights connecting neurons i and j in consecutive layers (from $l-1$ to l) are $w_{ij}^{(l)}$, and the bias term on the j^{th} neuron in the l^{th} layer is $b_j^{(l)}$. Each layer calculates an output based on the information received from the previous layer, as well as the optimized weights, biases, and the nonlinear activation function g_l (some examples include hyperbolic tangent sigmoid transfer function $g(z) = \frac{2}{1+e^{-2z}} - 1$, and softmax function $g(z_j) = \frac{e^{z_j}}{\sum_{i=1}^{H} e^{z_i}}$ where H is the number of class labels). Various common activation functions including ReLu, sigmoid, radial basis functions were presented and their performances were analyzed in [139]. In the output layer, y_{pred} is a vector giving the predicted probabilities of each class label. The predicted class label for the examined sample is indicated by the neuron with the highest probability, which in turn provides information on either the presence of a cyber-attack, or the type of the cyber-attack, depending on the classification problem the neural network is trained to solve.

To obtain an optimal set of weights and biases in Eq. 8.12, the Levenberg-Marquardt algorithm [81, 92] is used to minimize a Bayesian regularized mean squared error cost function, which has the following form:

$$S(w) = \mu \sum_{k=1}^{N_s} (y_{pred,k} - y_{true,k})^2 + \zeta \sum_{p=1}^{N_w} w_p^2 \tag{8.13}$$

where $k = 1, ..., N_s$ represents the number of samples in the training dataset, $p = 1, ..., N_w$ represents the number of weights and biases in the neural network, y_{true} is the vector of target class labels of each sample, y_{pred} is the vector of the predicted probabilities associated with each class label, and μ and ζ are the regularization hyper-parameters. Within the Levenberg-Marquardt algorithm, the gradient and the Hessian matrix of $S(w)$ are calculated using the backpropagation method. The weights and the data are assumed to have Gaussian prior probability distributions. Then, the regularization hyper-parameters, μ and ζ, are updated by maximizing their posterior probability distribution provided the data, which is equivalent to maximizing the likelihood of evidence by Bayes' Theorem. Within each epoch, two sequential procedures are carried out: the cost function $S(w)$ is minimized with respect to w, and the likelihood of evidence is maximized with respect to μ and ζ. Detailed formulation of this procedure can be found in [25]. Training and testing accuracies are calculated, which are the ratios between the number of correctly classified samples and total number of samples in the training and testing sets, respectively.

To develop an NN detector, state measurement data are collected while the system is operated under feedback controllers, i.e., the LMPC of Eq. 8.4 or the LEMPC of Eq. 8.5. For better detection accuracy, various state evolutions within the stability region under different operating conditions need to be accounted for; therefore, training data is collected for a broad range of initial conditions within the stability region Ω_ρ. Full state measurements $\bar{x}(t)$ are recorded along the time-varying trajectory for $t \in [t_0, t_{N_T}]$, and a nonlinear function denoted by $p(\bar{x})$ is computed. In order to provide an effective one-dimensional input feature for the detection problem, the function $p(\bar{x})$ needs to capture the dynamic behavior of all states. The selection of this input variable, $p(\bar{x})$, will be discussed in Section 8.3.1.

After data collection and adequate training, the NN detector is implemented online with the process controlled by MPCs with cyber-attack resilient control strategies that will be discussed in Section 8.4. The feed-forward NN model is a static model receiving inputs of fixed dimension,

N_T, which is the length of the time-varying trajectory. Therefore, the detection window of the NN detector while implemented online also matches the trajectory length of the training data, N_T. For example, the detector can be activated every time full state measurements become available, and uses a moving horizon detection window, receiving latest sequences of $x(t_k)$ of fixed length N_T. Alternatively, the NN detector can be activated at the end of each material constraint period (under EMPC), where $N_T = N_p$. In this case, the detector will receive the entire sequence of full state measurements $\bar{x}(t_k)$ over the latest material constraint period with a fixed length N_T. Each sample consists of a two-dimensional matrix $n \times N_T$, where n is the full state dimension, and N_T is the length of each state trajectory within the detection window. Each training sample corresponds to a different set of initial conditions for the closed-loop system simulation, and equal number of samples within each class labels are collected to ensure training accuracy.

8.3.1 Choice of Detection Input Variable

We first consider the case of LMPC. Since the control objective of the LMPC of Eq. 8.4 is to stabilize the system at the origin, for any initial condition in the operating region Ω_ρ, the closed-loop state profiles ultimately converge to their steady-state values if no attacks occur. Therefore, the closed-loop state profiles provide a good measure of system dynamic operations under LMPC, and thus, can be directly used as the NN input. However, unlike the case of operation under tracking MPC where the Lyapunov function decreases as the process states are driven towards the origin, off steady-state operation of LEMPC results in a state trajectory that remains on the boundary of the operating region Ω_ρ where $V(\bar{x}) = \rho$ to maximize process economic benefits. Considering this, the exact trajectory of each individual state variable is not predictable and does not follow a general expected trend even under nominal operation. Therefore, assessing the trajectory of the measured state vector might not be an effective method of detecting the occurrence of a cyber-attack in EMPC systems. Moreover, if the goal of a cyber-attack is to destabilize the closed-loop system within the shortest amount of time, the attacker will choose to set the current state measurement to the maximum/minimum allowable attack value characterized

by the boundary of the operating region Ω_ρ such that no alarms will be triggered. As a result, the falsified sensor measurements will also yield a Lyapunov function that is equal to ρ. The trajectory of the Lyapunov function $V(\bar{x})$ under nominal operation and under cyber-attacks can be too similar to differentiate. For these reasons, the Lyapunov function of the full-state measurements $V(\bar{x})$, which is used as an input variable for the detection algorithm used together with LMPC, is no longer a good measure of input for the detection algorithm when the system is operated under LEMPC.

Given that EMPC optimizes the economic benefit in its cost function, the progression of economic benefit is a measure that effectively reflects the time-varying operation under LEMPC; hence, information derived from the economic benefit provides a good comparison for attacked and not-attacked scenarios. Therefore, we will be monitoring the evolution of economic benefits during closed-loop operation. The cumulative economic benefit increases monotonically as operation time progresses. The first derivative of cumulative economic benefit (i.e., incremental economic benefit, which can be analogous to the reaction rate of desired product, at each sampling period) displays varying patterns depending on the initial conditions and on the material consumption constraint. The rate of change in the incremental economic benefit, or the change in the production reaction rate between sampling periods, provides information on the rate of change in the optimized cost function l_e inside the integral in Eq. 8.5a. This rate of change, which is also the second derivative of the cumulative economic benefit, will be used as the input parameters $p(\bar{x})$ for the neural-network-based detection algorithm.

8.3.2 Sliding Detection Window

As the NN detector may not have perfect classification accuracy, false alarms may occur based on a one-time detection where large oscillatory data within normal ranges may be misclassified as a cyber-attack. To reduce false alarm rates, a sliding alarm verification window is also implemented, where the number of positive attack detections within this window need to surpass a threshold before a cyber-attack alarm is confirmed. Specifically, a detection indicator D_i generated by each

sub-model M_i and a sliding detection window with length N_s are developed as follows:

$$D_i = \begin{cases} 1, & \text{if attack is detected by } M_i \\ 0, & \text{if no attack is detected by } M_i \end{cases} \tag{8.14}$$

Based on the detection indicator D_i at every N_a sampling steps, the weighted sum of detection indicators within the sliding detection window D_I shown in Fig. 8.2 at $t = t_k = k\Delta$ is calculated as follows:

$$D_I = \sum_{j=\lceil (k-N_s+1)/N_a \rceil}^{\lfloor k/N_a \rfloor} \lambda^{\lfloor \frac{k}{N_a} \rfloor - j} D_j \tag{8.15}$$

where λ is a detection factor that gives more weight to recent detections within the sliding window because the classification accuracy of the NN increases as more data is used for training. If $D_I \geq D_{TH}$, where D_{TH} is a threshold for the sliding alarm verification window, then the cyber-attack is confirmed and reported by the NN-based detection system; otherwise, the detection system remains silent and the sliding window will be rolled one sampling time forward. To balance false alarms and missed detections, the threshold D_{TH} is determined via extensive closed-loop simulations under cyber-attacks to derive a desired detection rate.

Additionally, since there is no guaranteed feasible control action that can drive the state back towards the origin once the state of the system of Eq. 8.1 is outside the stability region Ω_ρ, it is also necessary to check whether the state is in Ω_ρ, especially when cyber-attacks occur but have not been detected yet. Therefore, to prevent the system state from entering a region in state-space where closed-loop stability is not guaranteed, the boundedness of the state vector within the stability region can also be checked using the state measurement from redundant, secure sensors at the time when $D_i = 1$. If the state x has already left Ω_ρ, closed-loop stability is no longer guaranteed and in this case further safety system components (e.g., physical safety devices) need to be activated to avoid dangerous operations [191]. However, if $x \in \Omega_\rho$, the state measurement will be read from redundant, secure sensors instead of the original sensors to avoid deterioration of stability under the potential cyber-attack indicated by $D_i = 1$.

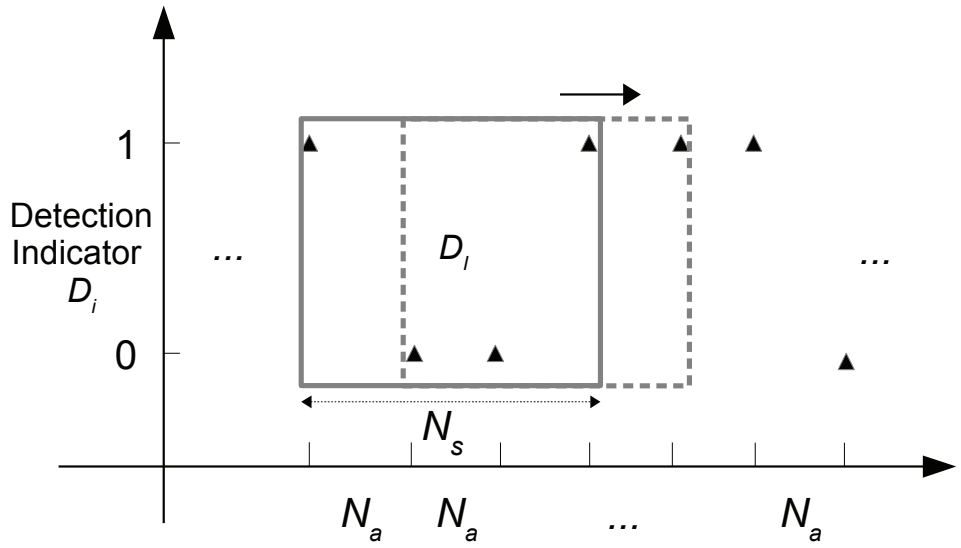


Figure 8.2: The sliding detection window with detection activated every N_a sampling steps, where triangles represent the detection indicator D_i and the box with length N_s represents the sliding detection window.

Remark 8.1. *The sliding window with length N_s is employed to reduce false alarm rates. Considering that the classification accuracy derived is not perfect, the idea behind the sliding detection window is that a cyber-attack is confirmed only if it has been detected for a few times continuously instead of a one-time detection. The length of sliding window N_s will balance the efficiency of detection and false alarm rates. Specifically, a larger N_s and a higher detection threshold D_{TH} ($D_I \geq D_{TH}$ within the sliding detection window represents the confirmation of a cyber-attack) lead to longer detection time but a lower false alarm rate, while a smaller N_s and a lower D_{TH} have the opposite effect. Therefore, N_s and D_{TH} should be determined well to achieve a balanced performance between detection efficiency and false alarm rate.*

Remark 8.2. *The above supervised learning-based cyber-attack detection method is able to distinguish the normal operation of the system of Eq. 8.1 from the abnormal operation under cyber-attacks, provided that there is a large amount of labeled data available for training. However, for those unknown cyber-attacks which are never used for training, the detection is not guaranteed. Specifically, if there exists an unknown cyber-attack that is distinct from the trained cyber-attacks, the NN-based detection method may not be able to identify it as a cyber-attack.*

In this case, an unsupervised learning-based detection method may achieve better performance by clustering unknown cyber-attack data into a new class. However, if the unknown cyber-attack shares similar properties (e.g., similar attack mechanism) with a trained cyber-attack, the NN method may still be able to detect it and classify it as one of the available classes.

8.4 Cyber-Attack Resilient Control Systems

In this section, we focus on the development of cyber-attack resilient control systems that can mitigate the impact of cyber-attacks upon detection. Several resilient control strategies are discussed for the closed-loop system of Eq. 8.1 under LMPC and LEMPC.

8.4.1 Redundant Sensors

When the cyber-attack is detected by $D_i = 1$ but not confirmed by $D_I \geq D_{TH}$ yet, the LMPC (LEMPC) optimization problem can use the state measurement from redundant, secure sensors instead of the original sensors as the initial condition $x(t_k)$ for the optimization problem of Eq. 8.4 (Eq. 8.5) until the next instance of detection. However, if the cyber-attack is finally confirmed by $D_I \geq D_{TH}$, the misbehaving sensor will be isolated, and the LMPC (LEMPC) optimization problem starts to use the state measurement from secure sensors instead of the compromised state measurement as the initial condition $x(t_k)$ for the optimization problem of Eq. 8.4 (Eq. 8.5) for the remaining time of process operation. The structure of the integrated cyber-attack-detection-control system for LMPC is shown in Fig. 8.3. If the cyber-attack is detected and confirmed before the closed-loop state is driven out of the stability region, it follows that the closed-loop state is always bounded in the stability region Ω_ρ thereafter and ultimately converges to a small neighborhood $\Omega_{\rho_{min}}$ around the origin for any $x_0 \in \Omega_\rho$ under the LMPC of Eq. 8.4. An example trajectory is shown in Fig. 8.4, where it is demonstrated that starting from an initial condition in Ω_ρ, the trajectory first moves away from the origin due to cyber-attacks and finally re-converges to a small neighborhood $\Omega_{\rho_{min}}$ around the origin under LMPC once the cyber-attack is detected by

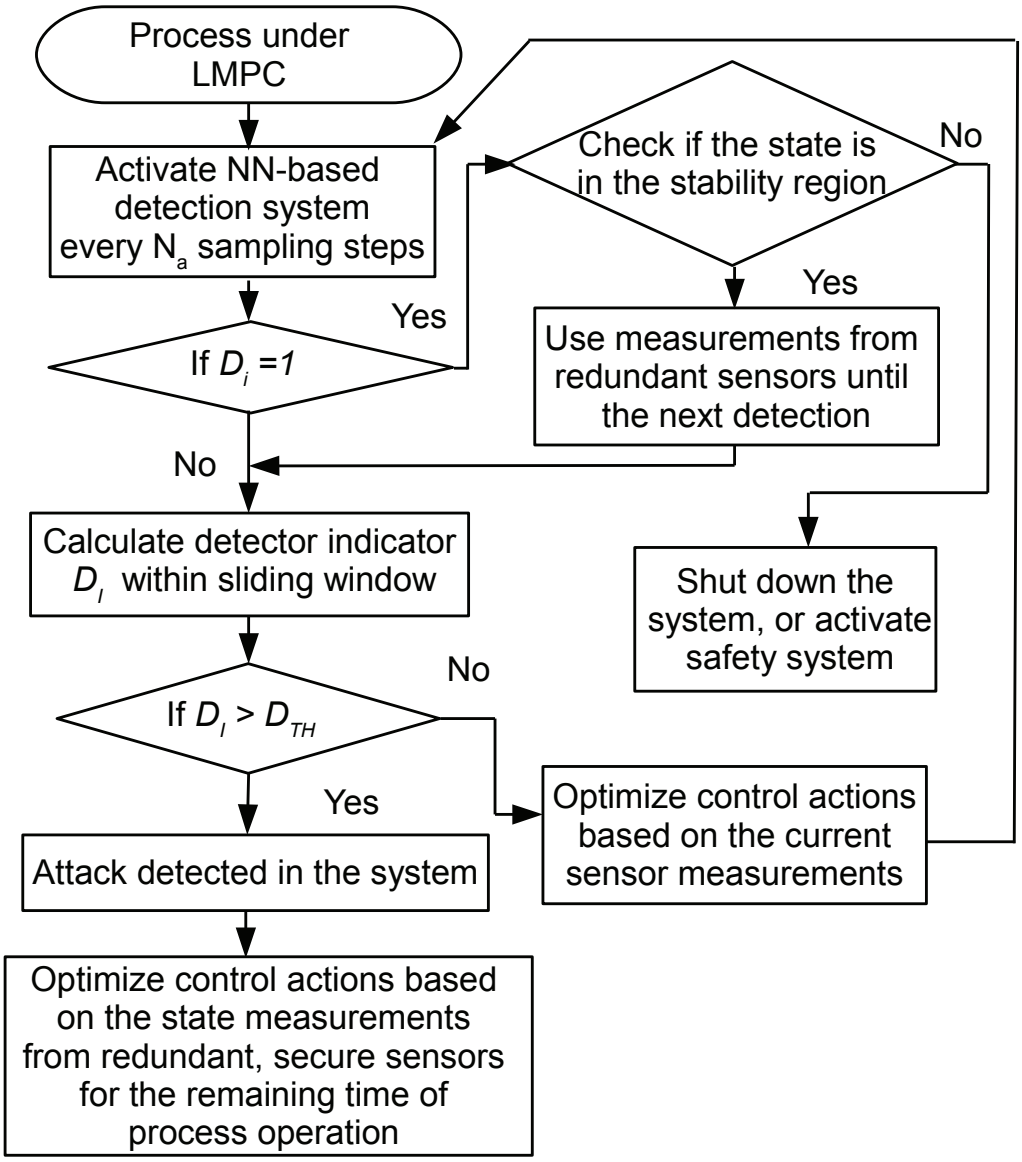


Figure 8.3: Basic structure of the proposed integrated NN-based detection and LMPC control method.

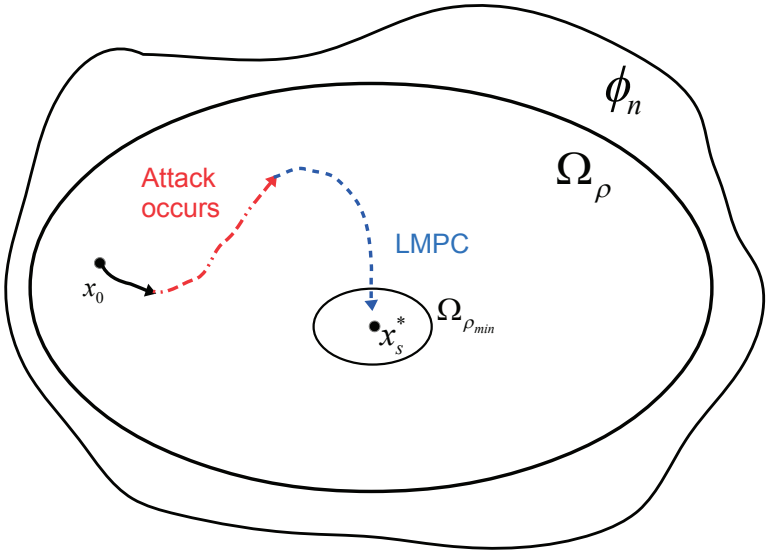


Figure 8.4: A schematic showing an example state trajectory under the integrated detection and control scheme.

the proposed NN-based detection scheme.

8.4.2 Attack-Resilient Combined Open-loop and Closed-loop Control

Upon the successful detection of cyber-attacks in sensors, one strategy that we have shown in Section 8.4.1 is to utilize the response plan that involves physical replacements of problematic sensors with their redundant back-up sensors. While sensor device replacement is an effective measure, there may be circumstances where redundant sensors cannot be deployed immediately, during which time the process may need to be operated in open-loop without reliable feedback measurements. Specifically, in this section, we consider the case of LEMPC, under which the state of the nominal system of Eq. 8.1 (i.e., no disturbances or cyber-attack) is bounded in Ω_ρ for all times. Additionally, since the LEMPC operates the system at the boundary of the operating region for the majority of operating time, we define a smaller level set $\Omega_{\rho_{secure}} := \{x \in \Omega_\rho \mid V(x) \leq \rho_{secure}\}$ inside the the stability region Ω_ρ as the new operating region such that the state may leave $\Omega_{\rho_{secure}}$ due to cyber-attacks but still remains in Ω_ρ before detection. Specifically, as the economic benefit of the process is maximized with respect to the state vector, it is likely that during the operating period, the optimized states will reach, and evolve along the boundary of the secure region $\Omega_{\rho_{secure}}$.

Assuming that the attacker has knowledge on the stability region as well as the secure region that the LEMPC operates based on, in order to induce maximum destructive impact on the system (e.g., in a min-max or surge cyber-attack) without triggering any alarms, the tampered state measurements will be near or on the boundary of the secure region $\Omega_{\rho_{secure}}$. Therefore, regardless of the presence of a cyber-attack, the measured process states will likely reach the boundary of $\Omega_{\rho_{secure}}$ where $V(\bar{x}) = \rho_{secure}$ during the operation of one material constraint period. In other words, when measured process states yield $V(\bar{x}) = \rho_{secure}$, there could be two reasons: 1) following optimized control actions $u^*(t_k)$, the measured process states reach the boundary of the bounded secure region $\Omega_{\rho_{secure}}$ at time t_k under the normal operation with no cyber-attacks, or 2) the measured states are compromised by a cyber-attack (e.g., min-max, or surge) at time t_k. Therefore, when measured states $\bar{x}(t_k)$ provide $V(\bar{x}(t_k)) = \rho_{secure}$, this measurement can no longer be trusted due to the ambiguous cause of this observation, and closed-loop control can no longer be carried out.

To combat the ambiguity of state measurements when they are on the boundary of $\Omega_{\rho_{secure}}$, open-loop control actions will be used in conjunction with closed-loop control. Assuming that the states measured at the beginning of each material constraint period, $t = t_{N_0}$, are secure and correct (the LEMPC can operate the system in multiple periods where the material constraint of Eq. 8.6 is satisfied in each operating period), the open-loop control actions are computed at the beginning of the material constraint period by solving the following nonlinear optimization problem:

$$\max_{u' \in S(\Delta)} \int_{t_{N_0}}^{t_{N_0+N_p}} l_e(\tilde{x}(t), u'(t))dt \tag{8.16a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t), u'(t)) \tag{8.16b}$$

$$u'(t) \in U, \ \forall \ t \in [t_{N_0}, t_{N_0+N_p}) \tag{8.16c}$$

$$\tilde{x}(t_{N_0}) = \bar{x}(t_{N_0}) \tag{8.16d}$$

$$V(\tilde{x}(t)) \leq \rho_{secure}, \ \forall \ t \in [t_{N_0}, t_{N_0+N_p}), \ \text{ if } \bar{x}(t_{N_0}) \in \Omega_{\rho_{secure}} \tag{8.16e}$$

$$\dot{V}(\bar{x}(t_{N_0}), u') \leq \dot{V}(\bar{x}(t_{N_0}), \Phi(\bar{x}(t_{N_0}))), \ \text{ if } \bar{x}(t_{N_0}) \in \Omega_{\rho} \backslash \Omega_{\rho_{secure}} \tag{8.16f}$$

where N_p is the number of sampling periods in one material constraint period, which is the

prediction horizon for open-loop control. At time t_k, a new material constraint period begins, the EMPC in open-loop control mode receives state measurement $x(t_k)$ and computes the optimal trajectory of N_p control actions that will be applied in a sample-and-hold manner until the end of this material constraint period. In the case that there are no cyber-attacks or process disturbances, this optimal trajectory of control actions would yield maximum economic benefits while meeting all input and state constraints.

While at closed-loop operation, if feedback measurement is no longer reliable and cannot be used for closed-loop control, the open-loop control actions that were calculated at the beginning of the material constraint period will be used as a substitute until the end of the material constraint period. At the end of the material constraint period, a cyber-attack detector is activated to determine any occurrence of an attack, and the reliability of the control system is re-assessed. The detector will provide information on the security status of the feedback measurements over the latest material constraint period. Upon mitigating the impact of a confirmed attack and/or confirming the security of the control system, closed-loop control with secure feedback measurement can be reactivated as a new material constraint period starts.

Although the absence of feedback may result in minor performance degradation in the case that process disturbances and modeling error exist and no cyber-attack is present, this strategy also completely eliminates the impact of a min-max or surge attack on the sensor measurements. The implementation strategy is illustrated in a logic flow diagram in Fig. 8.5, and the specific steps are outlined as follows:

1. At the start of a material constraint period ($t = t_{N_0}$), open-loop control actions over the course of the material constraint period are computed following Eq. 8.16. Closed-loop control is active, calculating the optimal control action over the next sampling period following Eq. 8.5 with $\Omega_{\rho_{secure}}$ replacing Ω_{ρ_e}.

2. If $\rho_{secure} - V(\bar{x}(t_k)) \leq c$, (where $c > 0$ quantifies the distance from the boundary of secure region to categorize a state measurement as being untrustworthy), then closed-loop control

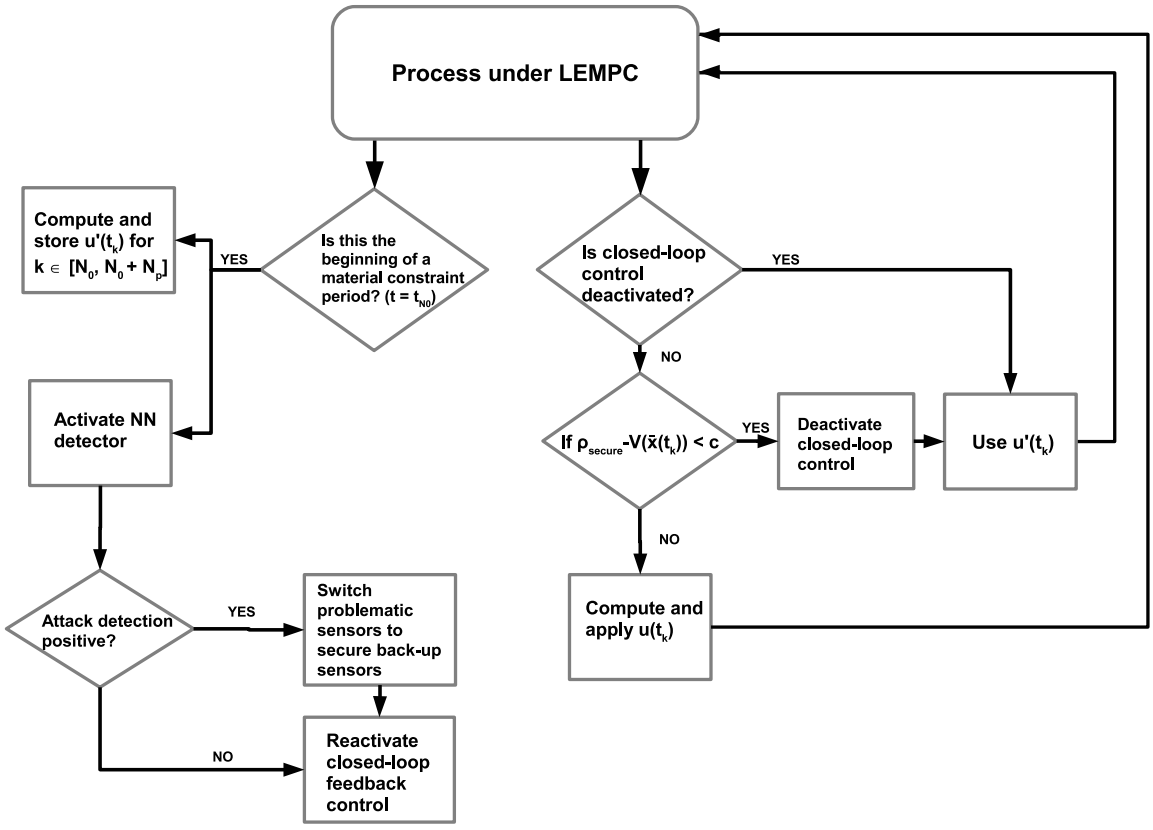


Figure 8.5: Logic flowchart outlining the implementation steps of the attack-resilient operation of LEMPC using combined closed-loop and open-loop control actions when operating within a secure region $\Omega_{\rho_{secure}}$.

(i.e., the LEMPC of Eq. 8.5) will be deactivated and open-loop control action $u'(t_k)$ calculated by the LEMPC of Eq. 8.16 will be used as an substitute.

3. Open-loop control actions $u'(t_k)$ will be used until $t_{N_0+N_p}$.

4. At $t_{N_0+N_p}$, the cyber-attack detector is activated to examine past full-state measurements $\bar{x}(t_k)$ for $k \in [N_0, N_0+N_p]$. If an attack is detected, then disconnect the tampered sensors, reroute these measurement signals to a set of secure back-up sensors, and go to Step 5. If detection indicates no attack, go to Step 5.

5. At $t_{N_0+N_p}$, a new material constraint period starts, and closed-loop control is reactivated.

Repeat Steps 1 – 4.

Remark 8.3. *In some cases, the system may never reach the boundary of* $\Omega_{\rho_{secure}}$ *depending on the initial condition, the size of* $\Omega_{\rho_{secure}}$*, and the length of the material constraint period. If this is the case, and cyber-attacks wrongfully set the measured states to be on the boundary of* $\Omega_{\rho_{secure}}$*, then closed-loop control will still be deactivated following the implementation of Step 2, and open-loop control actions will be used.*

8.4.3 Post Cyber-Attack State Reconstruction

In addition to redundant sensors and integrated open-loop and closed-loop control, in this section, we present a state reconstruction method to handle the compromised sensor measurements and continue process control following the successful detection of cyber-attacks. Measurement reconstruction has been of interest for many decades in the process fault detection field, e.g., [8, 54, 77, 127, 156, 160, 170]. As it is important to develop accurate detectors to promptly report the intrusion of a cyber-attack as well as building robust frameworks to mitigate the impact of cyber-attacks before the detector is activated, it is equally important to have recuperating measures in place to maintain controllability of the system in the absence of reliable sensors. The state reconstructor is developed to estimate the true state values using state measurements $\bar{x}$ and control actions u applied in real-time operation. In this section, we first introduce the recurrent neural network that is used to develop the state reconstructor using open-loop simulation data of the nonlinear system of Eq. 8.1. Subsequently, the state reconstructor is implemented in real-time to obtain estimated true state values based on closed-loop simulation data under attacks.

8.4.3.1 Recurrent neural network

Recurrent neural network (RNN) has been widely used in developing nonlinear dynamic functions based on time-series data to predict future states. The RNN structure is shown in Fig. 8.6 and its mathematical formulation can be found in Eq. 2.4. Since there exists a feedback loop in

its neurons, RNN models exhibit temporal behavior, and therefore, can be utilized to represent dynamic systems. The RNN-based state reconstructor is developed to estimate true state values in real-time based on faulty measurements $\bar{x}$ and control actions u. Specifically, the inputs to the RNN model are $\bar{x}(t)$ and $u(t)$, $\forall t \in [t_k, t_{k+r})$, where r is the number of sampling periods in the reconstruction window, and the output of the RNN models is the estimate of the true state x over $t \in [t_k, t_{k+r})$.

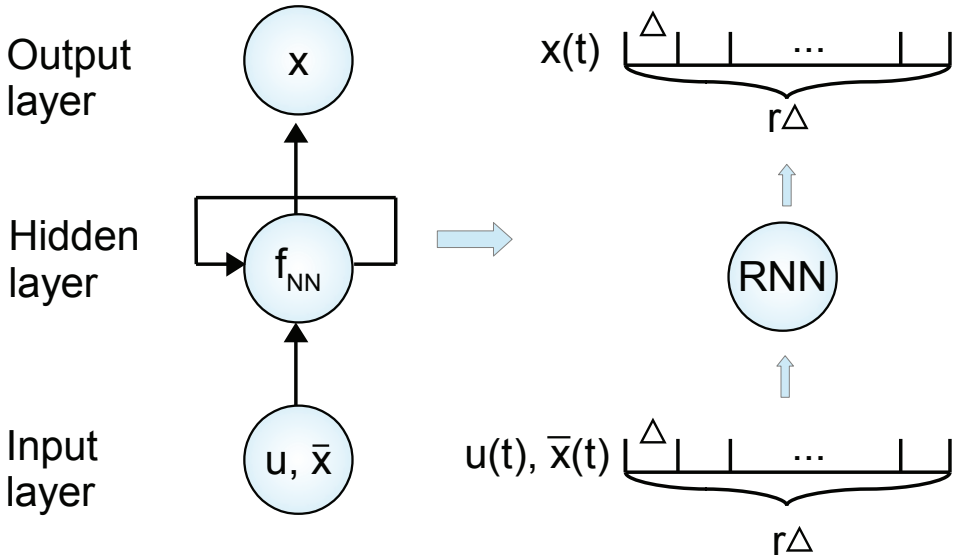


Figure 8.6: Recurrent neural network structure (left) and time-series input and output data (right), where $\bar{x}$, u are the input vectors, x is the output vector, Δ is the sampling period, $r\Delta$ is the length of reconstruction window of RNN model, and f_{NN} represents the hidden neurons that are used to capture the nonlinear relationship between input and output.

To develop RNN-based state reconstructors for the nonlinear system of Eq. 8.1 under the min-max, surge, and geometric cyber-attacks on sensor measurements that were introduced in the previous section, we first perform extensive open-loop simulations for the nonlinear system of Eq. 8.1 with $x \in \Omega_\rho$ and $u \in U$ under each of the different cyber-attacks, respectively. Specifically, starting from an initial condition $x_0 \in \Omega_\rho$, we apply a set of open-loop input sequences to the nonlinear system of Eq. 8.1 and introduce the above cyber-attacks at the second sampling period of each simulation run to obtain the trajectories of measured states and true states over a certain period of time (i.e., reconstruction window length $r\Delta$), respectively. Subsequently, the dataset that consists of extensive open-loop simulation runs is split into training, validation and testing

datasets, and the training process of RNN models is conducted following the standard procedure as introduced in Section 2.2.2 to minimize the difference between the predicted and the actual true state trajectories. Additionally, to ensure that the obtained RNN model can provide reliable state estimation for closed-loop operation of the nonlinear system of Eq. 8.1, the RNN model needs to be well trained such that the error between estimated states $\hat{x}$ and actual states x satisfies $|x - \hat{x}| \leq \gamma$, where $\gamma > 0$ is a sufficiently small bound. The RNN models are demonstrated to be able to capture the attacking patterns, for example, the zigzag pattern of measured states in the presence of min-max cyber-attack as shown in Fig. 8.11b, and provide the corresponding estimate of true state trajectory under a certain type of cyber-attack.

Remark 8.4. *The data-based state reconstruction approach can be applied in the closed-loop simulation of the nonlinear system of Eq. 8.1 provided that the sensor attacks are sparse attacks (i.e., a part of process state measurements remains secure), since the RNN model essentially generates the estimate of true states for those compromised sensors based on other secure sensor measurements and used control actions. Under the worst-case scenario that all the state measurements are under attacks, for example, the measured states remain unchanged for all times under attacks, it becomes barely possible for data-based state reconstructor to estimate the true states without any reliable information of secure sensors. In this case, an open-loop model-based control strategy could be applied to mitigate the impact of cyber-attacks to the greatest extent.*

Remark 8.5. *It is noted that since the dataset is generated using extensive open-loop simulations, the application of the RNN-based state reconstructor developed in this section is not restricted to the use of the LMPC of Eq. 8.4 or the LEMPC of Eq. 8.5. It can be applied to the closed-loop system of Eq. 8.1 using any other controller, for example, proportional-integral-derivative controller, provided that the state measurement is available at each sampling step. Therefore, the RNN-based state reconstruction provides a general approach to state estimation for the nonlinear system of Eq. 8.1 under sparse sensor attacks.*

8.4.3.2 Online reconstruction

Once cyber-attacks are detected by the NN-based detectors developed in Section 8.3, online state reconstruction will be implemented from the last secure checkpoint. Specifically, the RNN-based state reconstruction will be performed with the following steps. 1) Since the NN-based detectors can be implemented in real-time with a moving detection window to confirm the occurrence of cyber-attacks only if the cyber-attacks have been detected multiple times, the secure checkpoint will be set at the sampling step before the first detection to make sure the initial state measurement for the RNN reconstructor is not attacked. 2) Subsequently, the state reconstructor is applied to predict the state evolution from the last secure checkpoint to the current time step $t = t_k$ based on the sensor measurements and control actions in this period. Since the RNN model is developed with a reconstruction window length $r\Delta$, the estimated state in the second sampling period in the window will be used as the initial conditions for the next reconstruction as it moves one sampling step forward every time. 3) The estimated state $x(t_k)$ at the current time step will be sent to the controller (e.g., the LMPC of Eq. 8.4 or the LEMPC of Eq. 8.5) to calculate the control action $u(t_k)$ for the next sampling period $t \in [t_k, t_{k+1})$. 4) After the control action $u(t_k)$ is applied and the new state measurements $\bar{x}(t)$, $t \in [t_k, t_{k+1})$ are received, the state reconstruction window will be rolled one sampling time forward to estimate the true state value at $t = t_{k+1}$ using new compromised state measurements and control actions.

Remark 8.6. *It is noted that the RNN-based state reconstruction method is not restricted to the cyber-attacks discussed in this chapter since it is a data-driven approach that does not require any first-principles knowledge of process model or of cyber-attacks. For example, it can be applied to deception attacks such as optimization-based deception attack, randomly injected attacks and scheduled attacks on sensor measurements. However, there is one restriction, that is the cyber-attacks should target sensor measurements instead of blocking the communication networks between sensors and controllers, such that the RNN reconstructor can continuously receive (falsified) state measurements to make estimation. Therefore, the proposed approach may not be applied to cyber-attacks such as denial-of-service attack that makes sensor measurement*

unavailable to its intended users by temporarily disrupting network services.

8.4.3.3 Closed-loop control with reconstructed states

After the estimated state $\hat{x}(t_k)$ at the current time step $t = t_k$ is obtained through state reconstruction, the LMPC of Eq. 8.4 and the LEMPC of Eq. 8.5 will use the estimated state $\hat{x}$ instead of sensor measurement $\bar{x}$ to solve for the optimal control actions afterwards. However, considering that there may exist a state estimation error, in this section, we demonstrate that the RNN model needs to be well trained to achieve a desired estimation accuracy such that closed-loop stability is still guaranteed under LMPC/LEMPC with state reconstruction. The following proposition is developed to demonstrate that the error between true state trajectories x and the trajectories based on estimated states $\hat{x}$ of the nonlinear system of Eq. 8.1 is bounded under the same control actions for finite time.

Proposition 8.1. *Consider the solution $x(t)$ of the nominal system $\dot{x} = f(x,u,0)$ of Eq. 8.1 based on the actual state x, and the solution $\hat{x}(t)$ of the nonlinear system $\dot{\hat{x}} = f(\hat{x},u,0)$ based on the estimated state $\hat{x}$ with the initial condition $|x_0 - \hat{x}_0| \leq \gamma$, where $\gamma > 0$. If $x(t)$, $\hat{x}(t) \in \Omega_\rho$ for all times, then there exists a positive constant κ such that the following inequalities hold $\forall x, \hat{x} \in \Omega_\rho$:*

$$|x(t) - \hat{x}(t)| \leq \gamma e^{L_x t} \tag{8.17a}$$

$$V(x) \leq V(\hat{x}) + \alpha_4(\alpha_1^{-1}(\rho))|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{8.17b}$$

Proof. We define the state error vector as $e(t) = x(t) - \hat{x}(t)$ and derive the time-derivative of $e(t)$, $\forall x, \hat{x} \in \Omega_\rho$ and $u \in U$ using Eq. 8.3c as follows:

$$|\dot{e}| = |f(x,u,0) - f(\hat{x},u,0)| \leq L_x|e(t)| \tag{8.18}$$

Since the error between x_0 and $\hat{x}_0$ is bounded (i.e., $|x_0 - \hat{x}_0| \leq \gamma$), the upper bound for $|e(t)|$ is

derived for all $x(t), \hat{x}(t) \in \Omega_\rho$ as follows:

$$|e(t)| = |x(t) - \hat{x}(t)| \leq \gamma e^{L_x t} \tag{8.19}$$

Additionally, we derive Eq. 8.17b based on Eq. 8.2a, Eq. 8.2c and the Taylor series expansion of $V(x)$ around $\hat{x}$ for all $x, \hat{x} \in \Omega_\rho$ as follows:

$$\begin{aligned} V(x) &\leq V(\hat{x}) + \frac{\partial V(\hat{x})}{\partial x}|x - \hat{x}| + \kappa |x - \hat{x}|^2 \\ &\leq V(\hat{x}) + \alpha_4(\alpha_1^{-1}(\rho))|x - \hat{x}| + \kappa |x - \hat{x}|^2 \end{aligned} \tag{8.20}$$

where κ is a positive real number. □

The following proposition is developed to demonstrate that by implementing the stabilizing controller $u = \Phi(\hat{x}) \in U$ based on estimated states $\hat{x}$ in a sample-and-hold fashion after detection of cyber-attacks, $\dot{V}(x)$ for the nonlinear system of Eq. 8.1 can be rendered negative for all times such that the true state x can be driven towards the origin.

Proposition 8.2. *Consider the nominal system of Eq. 8.1 with $w(t) \equiv 0$ under the sample-and-hold implementation of the controller $u = \Phi(\hat{x}) \in U$ based on the estimated state $\hat{x}$ that satisfies $|\hat{x} - x| \leq \gamma$. Let $\varepsilon_s > 0$, $\Delta > 0$ and $\rho > \rho_s > 0$ satisfy*

$$-\alpha_3(\alpha_2^{-1}(\rho_s)) + L_x^{'}(\gamma + M\Delta) \leq -\varepsilon_s \tag{8.21}$$

Then, $\dot{V}(x) \leq -\varepsilon_s$ holds for any $x(t_k) \in \Omega_\rho \backslash \Omega_{\rho_s}$.

Proof. The time-derivative of $V(x(t_k))$ is obtained as follows:

$$\begin{aligned} \dot{V}(x(t_k)) =& \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), \Phi(\hat{x}(t_k)), 0) \\ =& \frac{\partial V(\hat{x}(t_k))}{\partial x} f(\hat{x}(t_k), \Phi(\hat{x}(t_k)), 0) + \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), \Phi(\hat{x}(t_k)), 0) \\ & - \frac{\partial V(\hat{x}(t_k))}{\partial x} f(\hat{x}(t_k), \Phi(\hat{x}(t_k)), 0) \end{aligned} \tag{8.22}$$

We can further derive the following inequalities using Eq. 8.2a, Eq. 8.2b and the Lip:cyberschitz condition of Eq. 8.3:

$$\begin{aligned}\dot{V}(x(t_k)) \leq & -\alpha_3(\alpha_2^{-1}(\rho_s)) + L_x^{'}|x(t_k) - \hat{x}(t_k)| \\ \leq & -\alpha_3(\alpha_2^{-1}(\rho_s)) + L_x^{'}\gamma\end{aligned} \tag{8.23}$$

Therefore, $\dot{V}(x) \leq -\varepsilon_s$ can be proved by further accounting for the impact of sample-and-hold implementation of control actions provided that Eq. 8.21 is satisfied as follows:

$$\begin{aligned}\dot{V}(x(t)) = & \frac{\partial V(x(t))}{\partial x} f(x(t), \Phi(\hat{x}(t_k)), 0) - \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), \Phi(\hat{x}(t_k)), 0) \\ & + \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), \Phi(\hat{x}(t_k)), 0) \\ \leq & L_x^{'}|x(t) - x(t_k)| + \dot{V}(x(t_k)) \\ \leq & L_x^{'} M\Delta - \alpha_3(\alpha_2^{-1}(\rho_s)) + L_x^{'}\gamma \\ \leq & -\varepsilon_s\end{aligned} \tag{8.24}$$

□

Based on the above proposition showing that $\dot{V}$ can be rendered negative within each sampling period, closed-loop stability for the nonlinear system of Eq. 8.1 under the LMPC of Eq. 8.4 can be readily proved, and therefore, is omitted here. The interested reader is referred to the similar proof for LMPC with secure state measurement in Section 2.3.2.

The next proposition demonstrates that Ω_{ρ_e} needs to be carefully chosen for the closed-loop system under LEMPC to ensure the invariance of the stability region Ω_ρ accounting for the estimation error.

Proposition 8.3. *Consider the nominal system of Eq. 8.1 with* $w(t) \equiv 0$ *under the sample-and-hold implementation of the LEMPC of Eq. 8.5. Let* $\Delta > 0$ *and* $\rho > \rho_e > \rho_s > 0$ *satisfy the following inequality:*

$$\rho_e \leq \rho - \alpha_4(\alpha_1^{-1}(\rho))\gamma e^{L_x\Delta} - \kappa(\gamma e^{L_x\Delta})^2 \tag{8.25}$$

If the state estimation error $|\hat{x}-x|$ is bounded by γ for all times, then, the true state of the nonlinear system of Eq. 8.1 under LEMPC is guaranteed to remain inside the stability region Ω_ρ, $\forall t \geq 0$, for any $x_0 \in \Omega_\rho$.

Proof. Following the results of Proposition 8.1, ρ_e is determined accounting for the error between true state trajectories x of the nonlinear system of Eq. 8.1 and the predicted trajectories based on estimated state $\hat{x}$ under the sample-and-hold implementation of control actions. The proof follows closely to that for LEMPC with secure state measurement in [57], and is omitted here. □

Therefore, given that the RNN model is well trained to achieve a sufficiently small estimation error, i.e., $|x-\hat{x}| \leq \gamma$, closed-loop stability is guaranteed for the nonlinear system of Eq. 8.1 under resilient LMPC and LEMPC using estimated state $\hat{x}$ upon detection of cyber-attacks.

Remark 8.7. *In this study, we assume no measurement noise, and thus, the RNN state reconstructor takes the compromised state measurement under cyber-attacks as the inputs to estimate the true state values. However, in the presence of measurement noise, which is very common in practical systems, the RNN reconstructor can still work well as long as the training dataset is developed from simulations/industrial process data that also account for the measurement noise with the same distribution. Additionally, closed-loop stability of MPC is still guaranteed provided that the modeling error of the RNN reconstructor is sufficiently small, which will be implemented as a constraint in the training process.*

8.5 Application to a Nonlinear Chemical Process

The application of the LEMPC of Eq. 8.5, the resilient control strategy presented in Section 8.4.2, as well as the training and online detection of NN cyber-attack detectors are demonstrated on the chemical reactor example that has been discussed in Chapter 7. Specifically, we consider an irreversible second-order reaction, $A \rightarrow B$, that transforms reactant A to product B at a reaction rate $r_B = k_0 e^{-E/RT} C_A^2$ in a well-mixed, non-isothermal continuous stirred tank reactor (CSTR). The

CSTR is equipped with a heating jacket that supplies or removes heat at a rate Q. The dynamic model of this CSTR process is described by the following material and energy balance equations:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 \tag{8.26a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{8.26b}$$

where the description of process variables can be found in Section 7.2.4, and a complete list of the process parameter values is given in Table 7.1. The CSTR is initially operated at the unstable steady-state $[C_{As},\ T_s] = [1.95\ kmol/m^3,\ 402\ K]$, and $[C_{A0_s}\ Q_s] = [4\ kmol/m^3,\ 0\ kJ/hr]$. The manipulated inputs are the inlet concentration of reactant A and the heat input rate represented by the deviation variables, i.e., $\Delta C_{A0} = C_{A0} - C_{A0_s}$ and $\Delta Q = Q - Q_s$, respectively. Additionally, considering the physical limitations, the manipulated inputs are bounded as follows: $|\Delta C_{A0}| \leq 3.5\ kmol/m^3$ and $|\Delta Q| \leq 5 \times 10^5\ kJ/hr$. Both the state and the inputs of the closed-loop CSTR system are represented in deviation variable forms, i.e., $x^T = [C_A - C_{As}\ T - T_s]$ and $u^T = [\Delta C_{A0}\ \Delta Q]$, respectively. Therefore, the equilibrium point of the system is at the origin of the state-space, (i.e., $x_s^T = [0,0], u_s^T = [0,0]$). We assume that at time $t = t_0$, the system is at the equilibrium point (i.e., the initial conditions of the system are $x_0 = [0,0]^T$).

The control objective of LEMPC is to maximize the economic profit of the CSTR process of Eq. 8.26 by manipulating the inlet concentration ΔC_{A0} and the heat input rate ΔQ, while maintaining the closed-loop state trajectories in the stability region Ω_ρ for all times. The objective function of the LEMPC optimizes the production rate of B as follows:

$$l_e(\tilde{x}, u) = r_B(C_A, T) = k_0 e^{-E/RT} C_A^2 \tag{8.27}$$

The dynamic model of Eq. 8.26 is numerically simulated using the explicit Euler method with an integration time step of $h_c = 2.5 \times 10^{-5}\ hr$. The nonlinear optimization problem of the LEMPC of Eq. 8.5 is solved using the MATLAB OPTI Toolbox with the sampling period $\Delta = 2.5 \times 10^{-3}\ hr$.

The LEMPC of Eq. 8.5 uses the following material constraint to make the averaged reactant

material available within one operating period $t_{N_p} = 0.06\ hr$ to be its steady-state value, C_{A0s} (i.e., the averaged reactant material in deviation form, u_1, is equal to 0).

$$\frac{1}{t_{N_p}} \int_0^{t_{N_p}} u_1(\tau) d\tau = 0\ kmol/m^3 \tag{8.28}$$

The control Lyapunov function $V(x) = x^T P x$ is designed with the following positive definite P matrix:

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{8.29}$$

The closed-loop stability region Ω_ρ for the CSTR with $\rho = 320$ is characterized as a level set of Lyapunov function inside the region D, from which the origin can be rendered asymptotically stable under the controller $u = \Phi(x) \in U$. The secure operating region $\Omega_{\rho_{secure}}$ for the LEMPC in Eq. 8.5 is selected to have $\rho_{secure} = 90$. Specifically, the design of the secure operating region $\Omega_{\rho_{secure}}$ can be adjusted depending on system dynamics and desired threshold for economic benefits. If the process dynamics is very fast, then more room needs to be vacated between Ω_ρ and $\Omega_{\rho_{secure}}$ to accommodate for the fast changes in process states when under cyber-attacks. However, designing a conservative secure operating region $\Omega_{\rho_{secure}}$ is at the expense of compromising economic benefits, since the maximum economic gain under normal operation is bounded by $\Omega_{\rho_{secure}}$. Therefore, the determination of the size of $\Omega_{\rho_{secure}}$ comes from a balance between operational stability and economic performance.

Resilient Operation of LEMPC

With initial conditions $x_0 = [0, 0]^T$, the closed-loop operation of the CSTR process in Eq. 8.26 over one material constraint period t_{N_p} under the LEMPC in Eq. 8.5, and under the resilient control of LEMPC with combined open-loop and closed-loop control actions as described in Section 8.4.2 around the secure operating region $\Omega_{\rho_{secure}}$ are both carried out. Fig. 8.7 presents the state-space plot showing the trajectory of the measured process states using the LEMPC of Eq. 8.5 and using

the resilient LEMPC control strategy when the process is under no attack. The switching from using closed-loop to open-loop control actions happens at $t_s = 0.0175\ hr$. For $t_0 \leq t_k < t_s$, measured process states are well within the secure operating region $\Omega_{\rho_{secure}}$, and closed-loop control using the LEMPC of Eq. 8.5 is used with state feedback updates. The LEMPC of Eq. 8.5 is deactivated at $t_s = 0.0175\ hr$ when the measured process states first reach the boundary of the secure operating region, and can no longer be trustworthy as this may be a result of a cyber-attack, i.e., when $\rho_{secure} - V(\bar{x}(t_k)) \leq c$, where $c = 0.5$ for this case study. Therefore, for $t_s \leq t_k \leq t_{N_p}$, control actions $u'(t_k)$ from the open-loop optimization of Eq. 8.16 that are solved based on the initial condition x_0 will be applied.

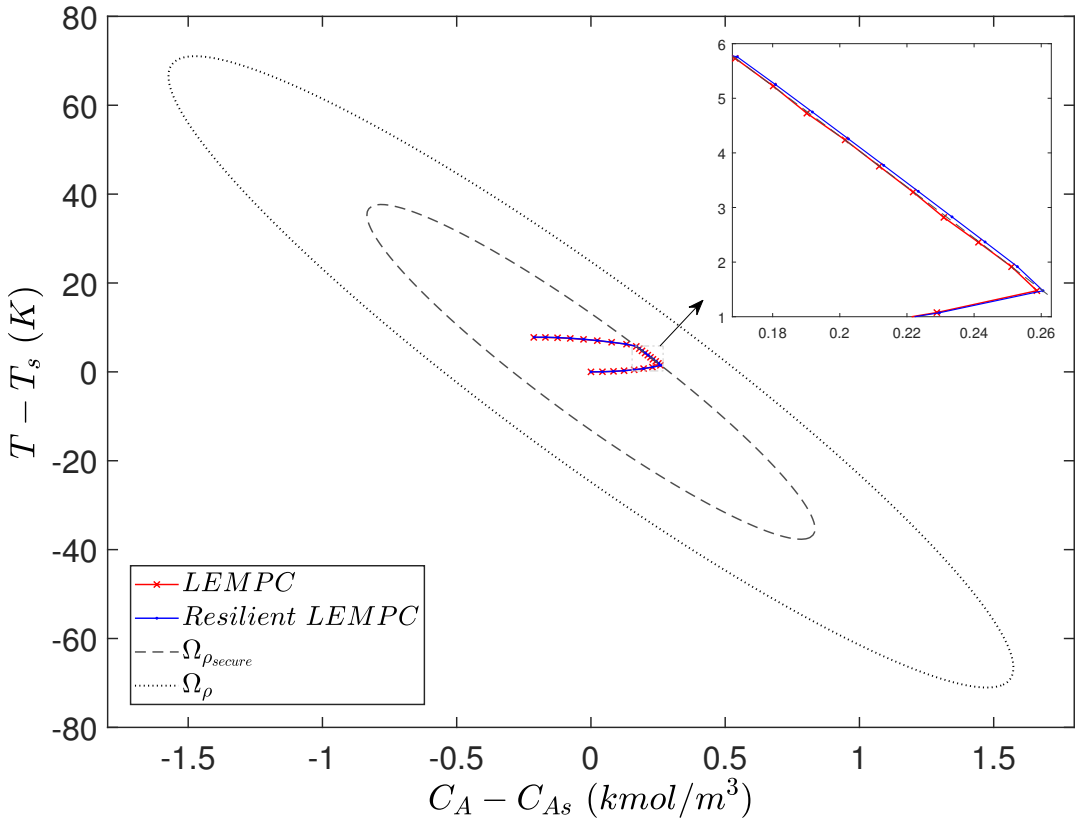


Figure 8.7: State-space plot showing the evolution of measured process states over one material constraint period under LEMPC (red trajectory) and under resilient LEMPC (blue trajectory).

Even in the case that no process disturbance, no model mismatch, and no cyber-attack is

present, the resulting state trajectories under LEMPC (closed-loop only), and the resilient LEMPC (closed-loop followed by open-loop control actions after the switching time t_s) are slightly different. This is because the prediction horizon used in the ordinary LEMPC with periodic closed-loop feedback has a length of $N = 8$ and rolls forward in time as feedback signal updates are received, whereas the open-loop optimization problem computed at the beginning of the material constraint period accounts for $N_p = 24$. Therefore, the control actions computed from the open-loop optimization, $u'(t_k)$, will be slightly different from $u(t_k)$ calculated from online optimization, resulting in slightly different state trajectories.

Despite the subtle differences in the state trajectory, using open-loop control actions following closed-loop control still maintains the process states within the secure operating region (hence the stability region) for all times. It is important to note that, if the process is operated at steady-state, the total economic benefits in the form of $\int_{t_0}^{t_{N_p}} l_e(\bar{x}(t))dt$ is 0.6397 $kmol/m^3$, which is much less than that achieved under time-varying EMPC operation. The total economic benefits from t_0 to t_{N_p} using closed-loop-only control actions from the LEMPC of Eq. 8.5 is 0.7936 $kmol/m^3$, and using the resilient control strategy outlined in Section 8.4.2 is similarly 0.7947 $kmol/m^3$. This shows the effectiveness of the resilient control strategy when the system is under no attack as it does not compromise system stability and economic performance. Furthermore, the similarity in the two trajectories also suggests that, if a cyber-attack is present and the resilient control strategy is utilized, the evolution of true process states will highly resemble that under closed-loop control in the absence of cyber-attacks.

Cyber-attack Resiliency Assessment

The purpose of using the resilient control strategy outlined in Section 8.4.2 is to prevent true process states from exiting the stability region Ω_ρ when under sensor cyber-attacks. Fig. 8.8 shows the state-space plot of the evolution of true process states and attacked state measurements from initial conditions $x_0 = [0, 0]^T$ over one material constraint period under LEMPC and under resilient LEMPC when the temperature sensor is attacked by min-max, geometric, replay and surge attacks,

respectively. In all cases, once the specified cyber-attack starts, it will continue until it has been successfully detected; the detection results and process simulation after the detection are shown in Section 8.5. Here, the simulation results over only one material constraint period are shown. After a cyber-attack has tampered the sensor, the resulting falsified state measurements will not exit the secure operating region $\Omega_{\rho_{secure}}$ so as to stay inconspicuous to the control engineer.

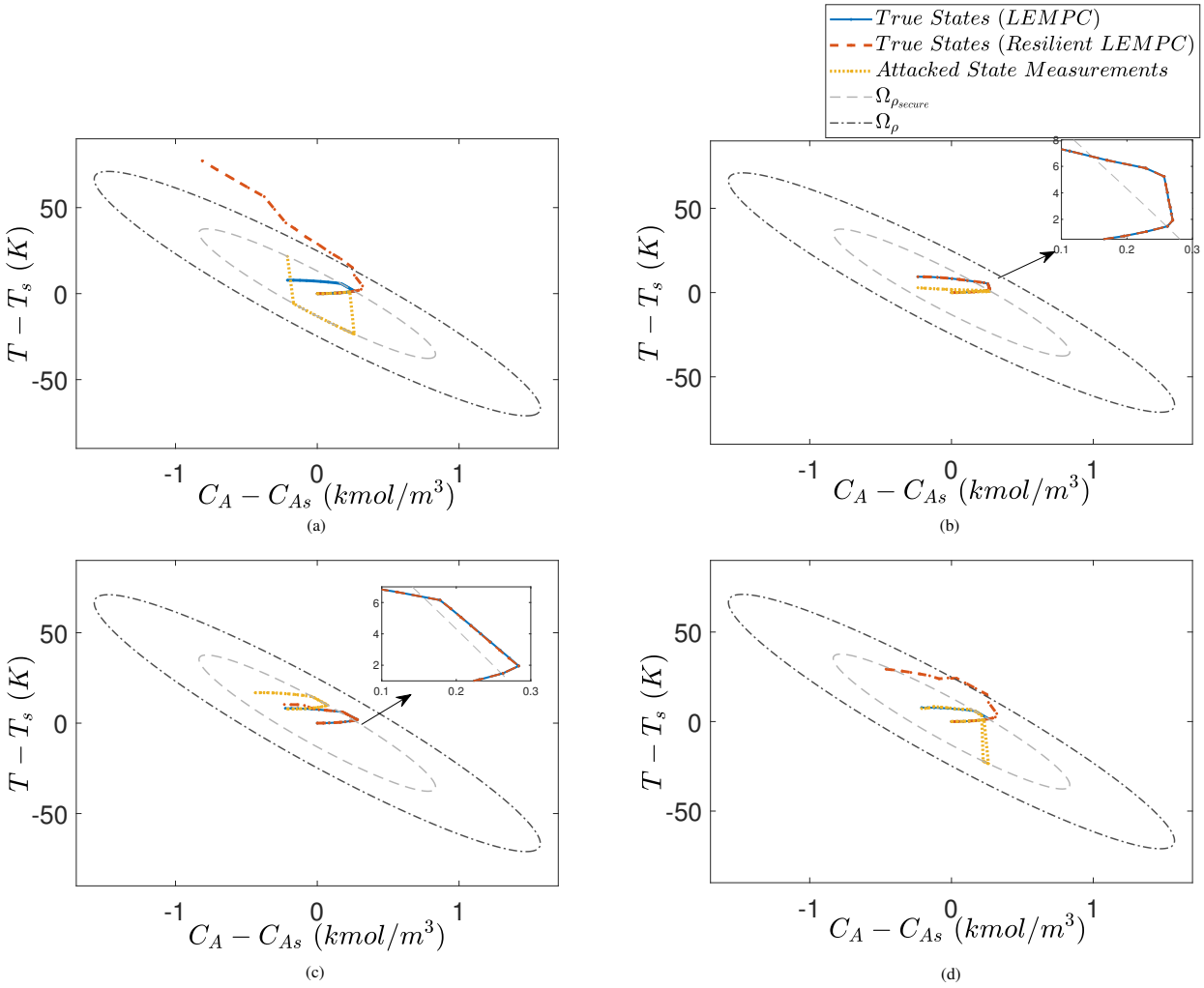


Figure 8.8: State-space plot showing the evolution of true process states and attacked state measurements (yellow trajectories) over one material constraint period under LEMPC (blue trajectories) and under resilient LEMPC (red trajectories) when (a) min-max, (b) geometric, (c) replay, and (d) surge attacks, are targeting the temperature sensor, where the dash-dotted ellipse is the stability region Ω_ρ and the dashed ellipse is $\Omega_{\rho_{secure}}$.

Min-max and surge cyber-attacks are added at $t = t_s = 0.0175\ hr$ such that there will be no

suspicious deviation in the Lyapunov function of the system. At $t = 0.0175\ hr$, both the true process state and the attacked state measurement will reach the boundary of the secure operating region, $V(x(t_s)) = V(\bar{x}(t_s)) = \rho_{secure}$. As shown in Fig. 8.8(a) and Fig. 8.8(d), when the temperature sensor is under min-max and surge attacks respectively, true process states will exit $\Omega_{\rho_{secure}}$ and eventually Ω_ρ if only closed-loop control actions from the online LEMPC optimization in Eq. 8.5 are used. However, when the resilient LEMPC control strategy is implemented, closed-loop control is deactivated at $t = 0.0175\ hr$, and the falsified feedback measurements can no longer impact the control system. Open-loop control actions, which are calculated based on a correctly measured set of initial conditions, are used starting at $t = 0.0175\ hr$ until the end of the material constraint period when $t = t_{N_p} = 0.06\ hr$. As a result, the true process states will not exit $\Omega_{\rho_{secure}}$, and the evolution of the true process states is almost identical to that under secure closed-loop control. The system stays resilient to min-max and surge attacks, with protected stability and comparable control performance.

However, the resilient control strategy may not be effective when the system is under other types of attacks, particularly in situations where the falsified state measurement does not approach the boundary of $\Omega_{\rho_{secure}}$. To illustrate this, geometric attacks on the temperature measurements as shown in Fig. 8.8(b) start at $t = 0.01\ hr$ following Eq. 8.8, where $\beta = x(t) * (1.001)$ and $\alpha = 0.1$. As cyber-attacks could happen at any time instant during operation, geometric attacks are designed and inserted as such to demonstrate the incapability of the resilient control strategy in handling geometric attacks or attacks alike. At $t = 0.01\ hr$, the states have not reached the boundary of $\Omega_{\rho_{secure}}$, therefore not satisfying the condition for deactivating closed-loop control. Geometric attacks starting at $t = 0.01\ hr$ resulted in state measurements that did not reach the boundary of $\Omega_{\rho_{secure}}$ for the entire duration of cyber-attack. Hence, closed-loop control continued with these false measurements, and the true process states exited $\Omega_{\rho_{secure}}$ during operation. Despite having a correct array of open-loop control actions computed at $t = 0\ hr$ using the correctly measured initial conditions, these control actions were not used. As a result, the resilient control strategy fails to ensure that the true process states are maintained within the secure operating region $\Omega_{\rho_{secure}}$.

Moreover, there may be situations where, even when closed-loop control is deactivated and feedback measurements are no longer used, the true process states still exit $\Omega_{\rho_{secure}}$ because the open-loop control actions are calculated based on false sensor measurements. To illustrate this scenario, replay attacks as shown in Fig. 8.8(c) start at $t_0 = 0\ hr$, and the replayed signals span the duration of one material constraint period. In other words, the replayed signals are real closed-loop state measurements when the system started from a different set of initial conditions, $\bar{x}_0 = [-0.2107\ kmol/m^3; 7.8047\ K]$. Since the initial conditions $\bar{x}_0$ are incorrect, open-loop control actions optimized over the prediction horizon of N_p based on $\bar{x}_0$ are also not correct. As a result, despite the falsified state measurements also reaching the boundary of $\Omega_{\rho_{secure}}$ at $t = 0.0175\ hr$ and deactivating closed-loop control, these incorrect open-loop control actions applied on the process still resulted in true process states exiting the secure operating region.

In this example, when under geometric and replay attacks, the true process states did not exit the stability region Ω_ρ; however, this may not be the case for a different geometric attack with larger α (geometric factor), a different replay attack that yielded more aggressive open-loop control actions, or for a faster process. In other words, system stability cannot be guaranteed by using the resilient control strategy, and an effective cyber-attack detection mechanism needs to be included.

Detectors Training and Testing

To train neural-network detectors, training data will be collected under closed-loop operation with the secure LEMPC outlined in Eq. 8.5. Simulation period is one material constraint period $t_{N_p} = 0.06\ hr$ with $N_p = 24$. Cyber-attacks are added at random times and last until the end of the simulation period. Neural network models are constructed and trained using the MATLAB Machine Learning and Deep Learning Toolboxes.

The reaction rate to yield product B, $r_B(\bar{x})$ can be calculated from full-state measurement $\bar{x}(t)$ at each time instant t_k from $k = 0$ to $k = N_p$ following Eq. 8.27, where $C_A = \bar{x}_1 + C_{A_s}$ and $T = \bar{x}_2 + T_s$. The input parameters used for neural network training are the time-varying trajectory of the rate of change in $r(\bar{x})$ over the simulation period of one material constraint period $N_p = 24$, which is

denoted as $p(\bar{x})$, shown as follows:

$$p(\bar{x}(t)) = \frac{dr(\bar{x})}{dt} \tag{8.30}$$

The evolution of $p(\bar{x})$ when the temperature sensor is under no attack, and under min-max, geometric, replay, and surge attacks, are shown in Fig. 8.9. Each sample consists of a 1×24 array of $p(\bar{x})$, started from a different initial condition within Ω_ρ. With extensive closed-loop simulations, equal number of samples are collected for each output label, from which 70% are used for training, and 30% are used for testing.

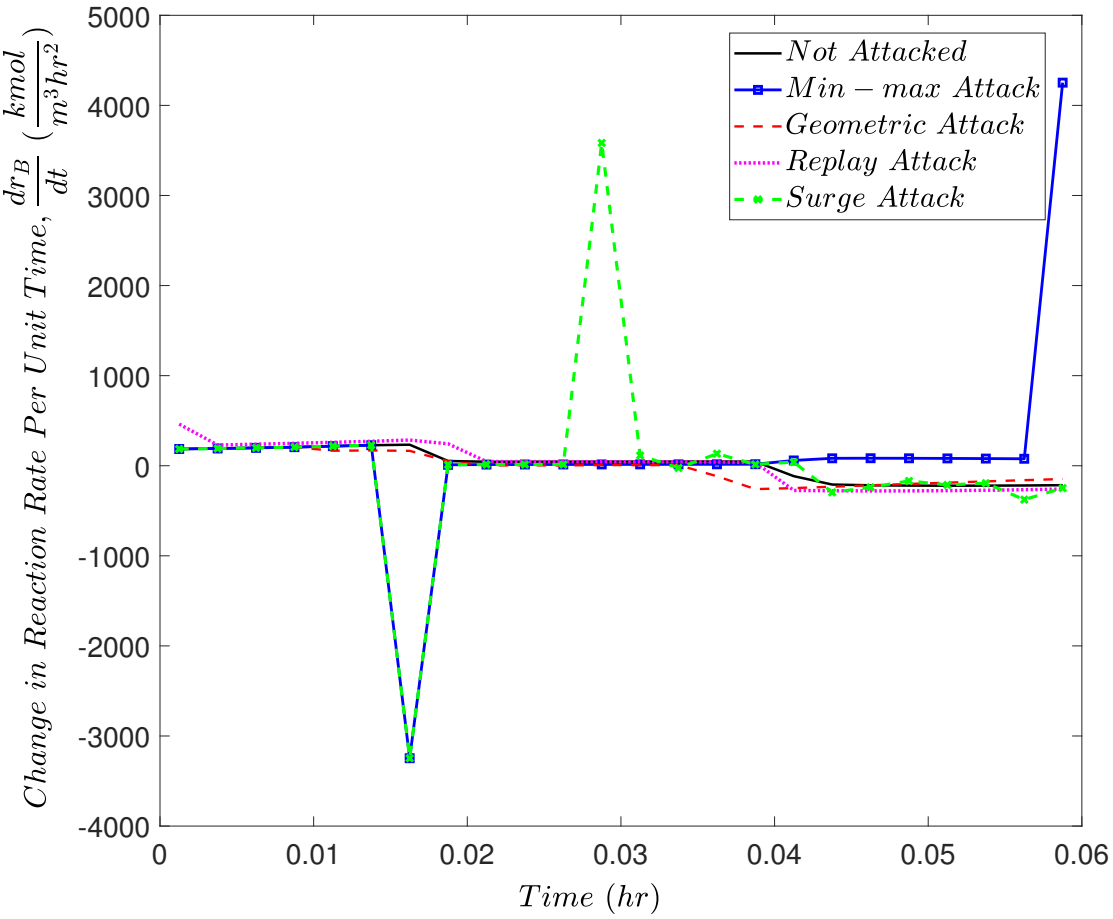


Figure 8.9: Time-derivative of the reaction rate r_B of Eq. 8.27 based on measured process states over one material constraint period, when the temperature sensor is under no attack, and under min-max, geometric, replay, and surge attacks, respectively.

First, min-max attacks are used to train a neural-network-based detector. This feed-forward

neural network model has two hidden layers with 12 and 10 neurons in each layer respectively. Both hidden layers use a *tansig* activation function, which is in the form $g_{1,2}(z) = \frac{2}{1+e^{-2z}} - 1$. The output layer uses a $softmax$ function to provide a predicted probability of the class labels, which is in the form of $g_3(z_j) = \frac{e^{z_j}}{\sum_{i=1}^{H} e^{z_i}}$ where H denotes the number of class labels. Bayesian regularized mean squared error cost function $S(w)$ are minimized with respect to the weights and biases using the Levenberg-Marquardt algorithm, in which the gradient and the Hessian matrix of $S(w)$ are calculated using the back-propagation method. A total of 750 samples are collected for each class label. The training time for this 2-class detector is 2.05 seconds, undergoing 70 epochs, and the detector achieves a training accuracy of 98.9%. The testing accuracy of this detector against the different attack types is shown in Table 8.1. Note that geometric attacks are not identified as being attacked due to the vast difference in the trends of $p(\bar{x})$ when under geometric attack compared to min-max attacks as shown in Fig. 8.9.

A second detector is trained with min-max and geometric attacks. The detector is able to classify between 3 classes: not attacked, attacked by min-max cyber-attacks, and attacked by geometric cyber-attacks. Thus, the detector is capable of differentiating the types of cyber-attacks in addition to indicating the presence of one. This detector is trained because geometric attacks exhibit very different behavior than min-max attacks, and therefore the testing accuracy by the 2-class detector is very low. This 3-class feed-forward neural network detector has two hidden layers with 15 and 12 neurons each, using the same activation functions and cost function in Eq. 8.12, which is minimized using the Levenberg-Marquardt algorithm. The training time for this 3-class detector is 39.48 seconds with 300 epochs. This 3-class detector achieves an overall training accuracy of 91.8%, and its testing accuracies in response to min-max, geometric, and surge attacks are shown in Table 8.1. The detector accurately identifies min-max and geometric attacks as their respective labels, and it classifies 71.0% of surge attacks as min-max, 10.0% as geometric, and the remaining 19% are wrongly classified as "not attacked".

Remark 8.8. *Since replay signals could mimic the secure operation of one entire material constraint period starting at a different initial condition, they are essentially a different sample that*

Table 8.1: Detection accuracies of NN detectors in response to min-max, geometric, and surge attacks.

	Detector 1 (Attacked vs. Not Attacked)	Detector 2 (Min-max vs. Geometric vs. Not Attacked)
Min-max	98.3%	89.7%
Geometric	2.4% (Attacked)	71.1%
Surge	87.0% (Attacked)	71.0% (Min-max); 10.0% (Geometric)
Not Attacked	98.4%	95.6%

belongs to the class of "not attacked", and will be rightfully classified as being "not attacked". At the end of the material constraint period, the falsified signals follow exactly the trajectory of previous secure measurements of one period, thus they will remain undetectable by the NN detectors.

Online Detection

Detector 1 is used to detect min-max and surge attacks, whereas detector 2 is used to detect geometric attacks. The corresponding detector is activated at the end of the material constraint period, and examines state measurements received over the last material constraint period. Since replay attacks cannot be detected, the online detection results are also not shown. Fig. 8.10 shows the evolution of true process states and measured process states attacked by min-max, geometric, and surge cyber-attacks when the process is controlled by the resilient LEMPC with combined open-loop and closed-loop control. The figures show the trajectories over two material constraint periods, where NN-based detection occurs twice – once at the end of the first period, and once at the end of the second period. Min-max and surge attacks are correctly detected by detector 1 at the end of the first constraint period $t = 0.06\ hr$ by examining the trajectory of $p(\bar{x}(t))$ from $t = 0\ hr$ to $t = 0.06\ hr$, after which the sensor devices are switched to a secure set of redundant sensors and operation continues with these secure sensor measurements. During the second period, the attacked old set of sensors are no longer connected to the control system, and the newly switched

set of sensors are not tampered by cyber-attacks. At the end of the second material constraint period $t = 0.12\ hr$, detector 1 is activated again, and it correctly classifies the secure measurements as "not attacked".

Furthermore, if a particular attack type is trained as a separate class (i.e., "geometric") from other attack types (i.e., "min-max"), then the detector is also capable of identifying the type of cyber-attack. As shown in Fig. 8.10(b), although the true process states exited $\Omega_{\rho_{secure}}$ during the first material constraint period (closed-loop control based on false feedback signals was not deactivated), the state measurements attacked by geometric attacks were still correctly identified as geometric by detector 2 at the end of the first material constraint period. After switching the sensor devices to the respective secure back-up sensors, detector 2 correctly identifies the trajectory of $p(\bar{x}(t))$ over the second material constraint period from $t = 0.06\ hr$ to $t = 0.12\ hr$ as "not attacked". This means that, although the resilient control strategy cannot ensure stability over one material constraint period if the attacked measurement deliberately avoids approaching the boundary of $\Omega_{\rho_{secure}}$, the attack can still be detected at the end of the material constraint period, and mitigation measures can be taken following the successful detection to terminate the impact of the cyber-attacks. Therefore, setting a shorter material constraint period in addition to operating within a conservative secure region could be another preventative method to consider, so that the cyber-attack detection can happen more frequently.

Real-time State Reconstruction

In addition to the integrated open-loop and closed-loop control, we also carry out the closed-loop simulations for the CSTR system of Eq. 8.26 under LEMPC with state reconstruction that was discussed in Section 8.4.3. In this case, we assume that the CSTR system of Eq. 8.26 is normally operated in the region Ω_{ρ_e} with $\rho_e = 280$ under no attacks, and the cyber-attack detection is implemented in real-time, i.e., at each sampling period, instead of after each material constraint period. When cyber-attacks occur, the true state trajectory may leave Ω_{ρ_e} under LEMPC, and therefore, the size of Ω_{ρ_e} is carefully chosen to maintain the state within the stability region Ω_ρ

before the detection of cyber-attacks.

Two-hidden-layer RNN models with 60 neurons in each layer are designed using the state-of-the-art machine learning library, Keras, to train the state reconstructors for min-max, surge, and geometric cyber-attacks, respectively with datasets consisting of around 150,000 data sequences. The averaged mean square errors of the three state reconstructors on training and validation datasets are maintained below 10^{-5}. The averaged training time for each neural network is around 2.5 hr. The training is done off-line, and the obtained RNN model is used on-line for state estimation within MPC. It is noted that the state estimation within MPC is completed almost instantaneously because the RNN model after training is essentially a nonlinear function that calculates estimated values (output) given the past state measurements (input). Therefore, the computational time for running estimation using RNN models is negligible compared to the process sampling time.

The closed-loop state trajectories and profiles for min-max, surge, and geometric cyber-attacks under LEMPC are shown in Fig. 8.11a-8.11b, Fig. 8.12a-8.12b, and Fig. 8.13a-8.13b, respectively. Specifically, in Fig. 8.11a, it is shown that starting from the initial condition $x_0 = (0,\ 0)$, the system of Eq. 8.26 is initially operated without any attacks. Then, the min-max cyber-attack is introduced on the temperature sensor at $t = 0.05$ hr, and it is shown that the sensor measurement (dashed red trajectory) stays on the lower boundary of Ω_{ρ_e}, while the true state trajectory (blue) starts exiting the Ω_{ρ_e} from the upper boundary. Once the cyber-attack is detected at $t = 0.07$ hr, we reconstruct the true states (colored dotted trajectories) based on past sensor measurements and control actions, and subsequently, the LEMPC of Eq. 8.5 restabilizes the CSTR system by using the estimated state. In Fig. 8.11b, it is demonstrated that the reconstructed concentration and temperature are very close to the true states in closed-loop simulation, and therefore, provide reliable state estimation for the feedback control with LEMPC. During online implementation, state reconstruction will be ideally activated after the first positive detection given by the cyber-attack detector to save computational power, given that detection happens in real-time and promptly reports the occurrence of a cyber-attack. However, starting state reconstruction is not limited to only when the detector

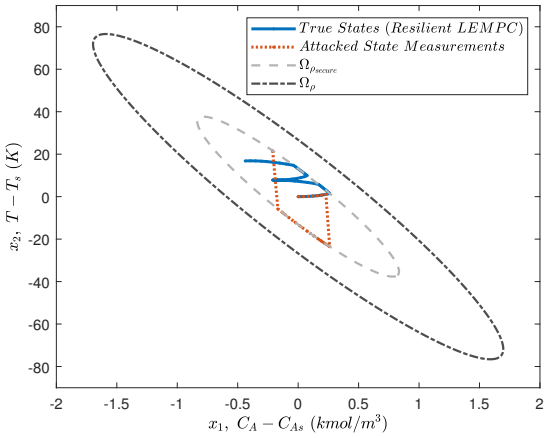
gives a positive detection. Here, we have plotted the reconstructed states right after the attack occurs to demonstrate the effectiveness of this RNN-based state reconstruction method throughout the attack duration. Moreover, even in the case that the sensor measurements are not faulty, the NN-based state reconstructor is also capable of predicting the true process states successfully with a sufficiently small bounded error. Therefore, state reconstruction could start at the beginning of the operation period, as long as the sensor measurements prior to which time are reliable.

In Fig. 8.12a-8.12b, we perform closed-loop simulation under surge cyber-attack for multiple EMPC operating periods. It is demonstrated in Fig. 8.12b that the surge cyber-attacks are introduced in each material constraint period (i.e., from $t = 0$ hr to $t = 0.15$ hr, from $t = 0.15$ hr to $t = 0.3$ hr, and from $t = 0.3$ hr to $t = 0.45$ hr with $t_{N_p} = 0.15$ hr), from which the compromised sensor measurement first reaches its maximum allowable value and remains a small deviation from true states afterwards. Similarly, RNN-based state reconstructor successfully estimates the true state trajectory and provides a reliable correction for sensor measurement for LEMPC. Additionally, Fig. 8.13a-8.13b show the simulation results of closed-loop CSTR system under geometric cyber-attack, for which the analysis is similar to the above, and is omitted here.

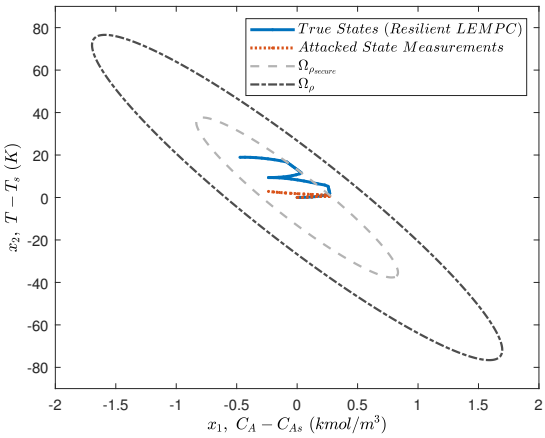
8.6 Conclusions

In this chapter, the secure operation of nonlinear chemical processes under MPC/EMPC was presented via the design of resilient control strategies, and a neural-network-based cyber-attack detector. Considering a general class of nonlinear systems, the NN-based detection system was first developed with the sliding detection window to detect intelligent cyber-attacks. Subsequently, resilient control systems were developed with several control strategies including redundant sensors, combined open-loop and closed-loop control, and post cyber-attack state reconstruction. Through simulating a continuously stirred tank reactor process, it was demonstrated that the proposed control strategy was effective in maintaining process stability against particular types of malicious cyber-attacks, namely min-max, geometric and surge attacks, while achieving

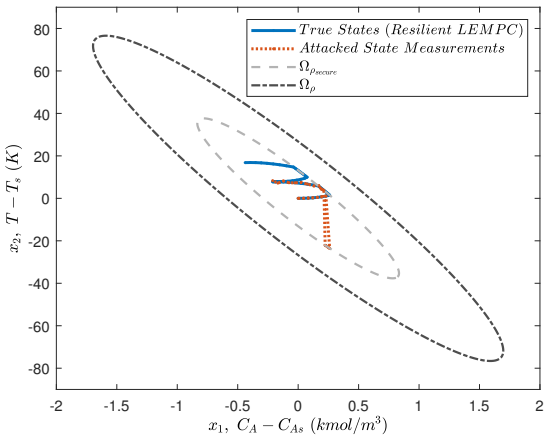
comparable economic performance compared to nominal operation under no attacks. A neural-network-based cyber-attack detector was able to provide a diagnosis at the end of each LEMPC operation period, and simulation results demonstrated that min-max and surge attacks could be successfully detected. Additionally, the RNN-based state reconstructor successfully estimated the true states in real-time implementation of LEMPC such that closed-loop stability of the nonlinear processes can be guaranteed upon cyber-attack detection.



(a)

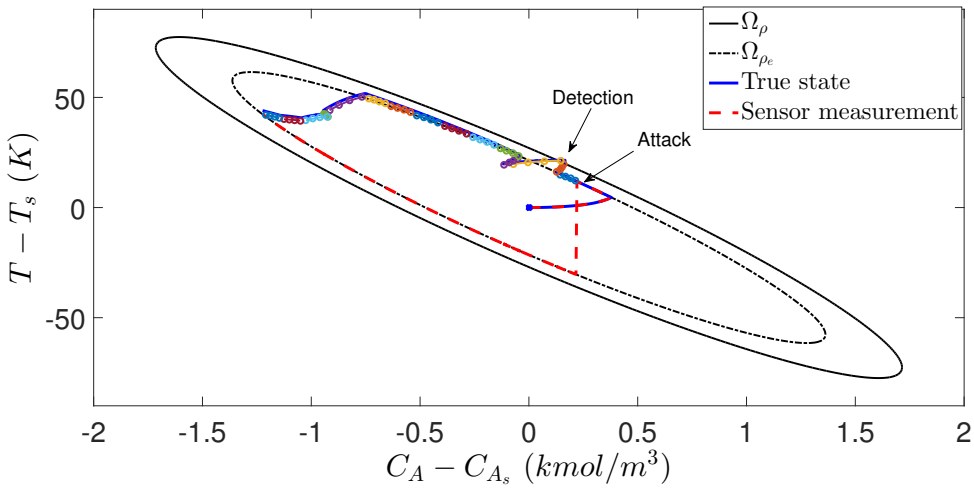


(b)

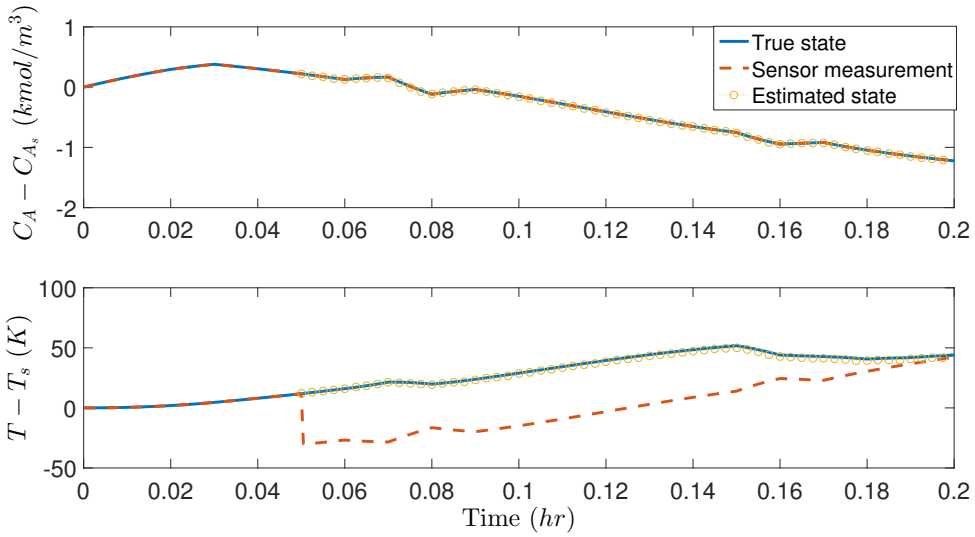


(c)

Figure 8.10: State-space plot showing the evolution of true process states (blue trajectories) and attacked state measurements (red trajectories) over two material constraint periods under the resilient LEMPC when (a) min-max, (b) geometric, and (c) surge attacks, targeting the temperature sensor are successfully detected by a NN detector at the end of the first material constraint period, $t = 0.06\,hr$, where the dash-dotted ellipse is the stability region Ω_ρ and the dashed ellipse is $\Omega_{\rho_{secure}}$.

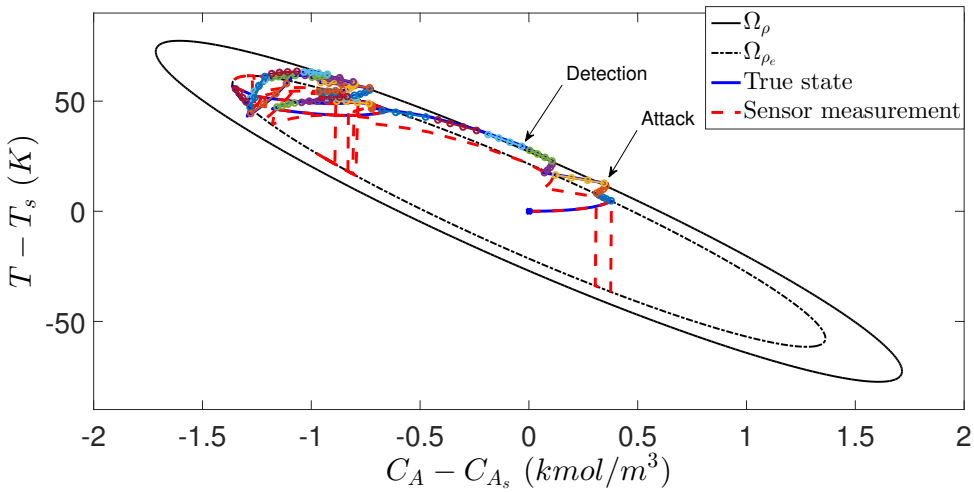


(a)

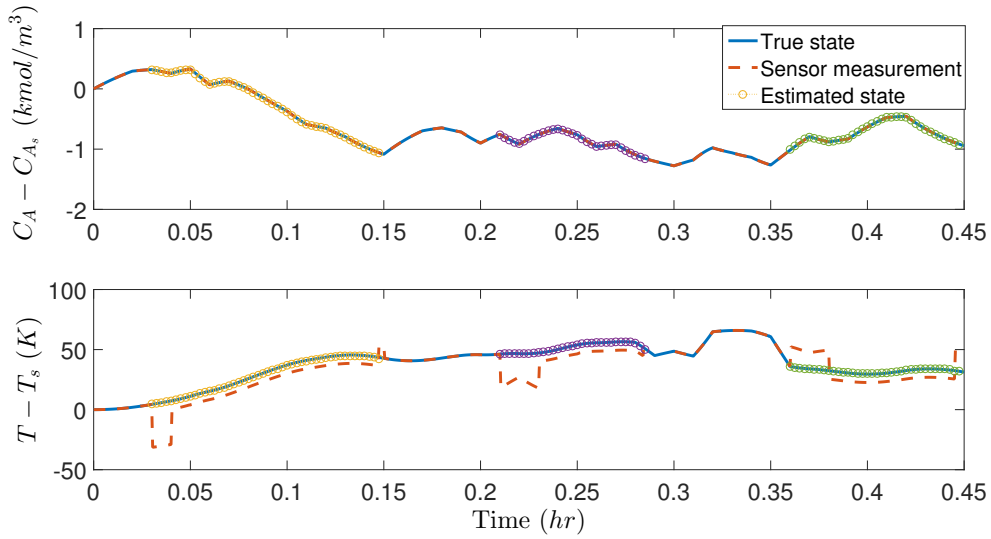


(b)

Figure 8.11: (a) State-space trajectories, and (b) closed-loop profiles of true state (blue), measured state (red), and reconstructed state (marked by colored circles) for the CSTR system of Eq. 8.26 under LEMPC when a min-max cyber-attack is introduced at $t = 0.05$ hr on the temperature sensor.

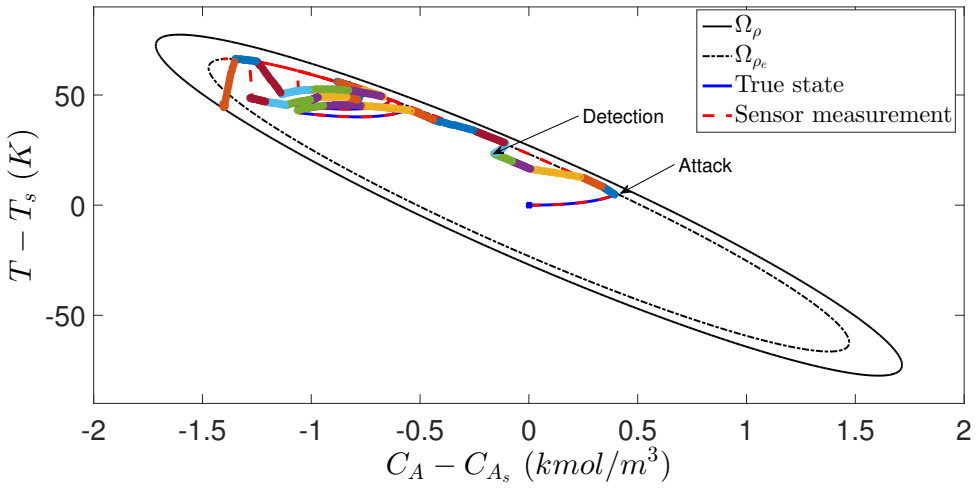


(a)

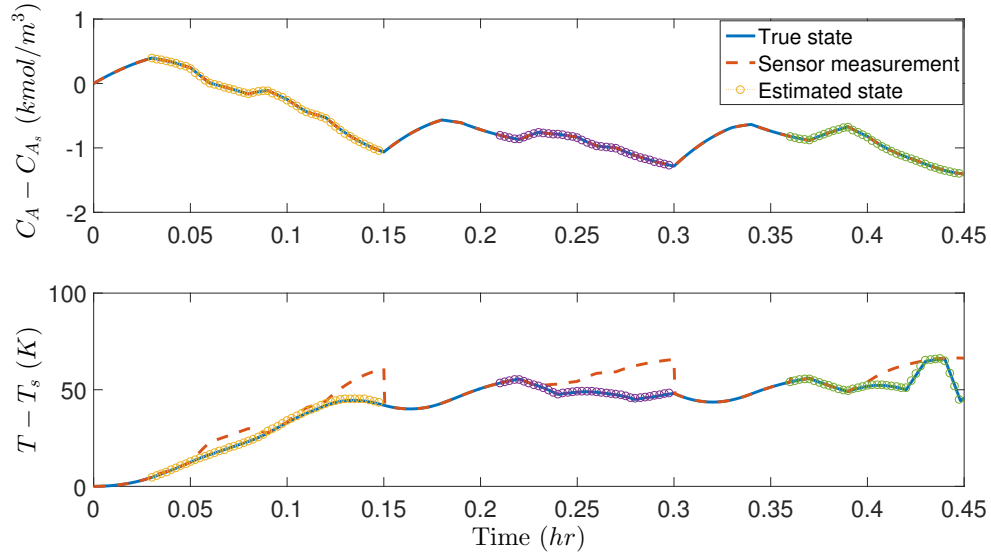


(b)

Figure 8.12: (a) State-space trajectories, and (b) closed-loop profiles of true state (blue), measured state (red), and reconstructed state (marked by colored circles) for the CSTR system of Eq. 8.26 under LEMPC when surge cyber-attacks are introduced at $t = 0.03$ hr, $t = 0.21$ hr, and $t = 0.36$ hr on the temperature sensor.



(a)



(b)

Figure 8.13: (a) State-space trajectories, and (b) closed-loop profiles of true state (blue), measured state (red), and reconstructed state (marked by colored circles) for the CSTR system of Eq. 8.26 under LEMPC when geometric cyber-attacks are introduced at $t = 0.03$ hr, $t = 0.21$ hr, and $t = 0.36$ hr on the temperature sensor.

Chapter 9

Conclusion

This dissertation provides a number of rigorous methods for the design of machine-learning-based MPC systems to improve process operational safety and cybersecurity for chemical processes described by nonlinear dynamic models. Specifically, data-driven models were developed for nonlinear dynamic processes using machine learning techniques, and then incorporated in MPC and EMPC schemes to predict process dynamics in solving the optimization problems. Following that, the real-time implementation of machine-learning-based MPC/EMPC with online learning of machine leaning models was discussed. Physics-based machine learning modeling methods were further presented to improve model performance by accounting for a priori process knowledge. Subsequently, model predictive control (MPC) and economic MPC schemes that use control Lyapunov-barrier functions (CLBF) to ensure closed-loop stability and operational safety were presented with rigorous stability and safety analysis. Then, the development of machine-learning-based CLBF-MPC and CLBF-EMPC schemes were presented with process stability and safety analysis. Finally, an integrated detection and control system for process cybersecurity was developed, in which several types of intelligent cyber-attacks, machine learning detection methods and resilient control strategies were presented.

In Chapters 2, 3, 4 and 5, machine learning techniques were utilized to develop data-driven models to approximate nonlinear dynamic processes. Specifically, in Chapters 2 and 3, the

concept of recurrent neural networks (RNN) and a general framework to develop RNN models for nonlinear dynamic systems were first introduced. Machine-learning-based MPC and EMPC schemes were developed by incorporating RNN model as the prediction model, with sufficient conditions provided to ensure closed-loop stability. Parallel computing of an ensemble of machine learning models were developed to improve computational efficiency in training and in closed-loop operation under RNN-MPC. In Chapter 4, online learning of machine learning models were utilized to update machine learning models using the most recent process data in order to improve model accuracy for the nonlinear process subject to time-varying disturbances. In Chapter 5, three physics-based machine learning modeling approaches (i.e., hybrid model, partially-connected RNN model, and weight-constrained RNN model) that incorporate a priori process knowledge into RNN models were developed to improve model accuracy. The effectiveness of all the aforementioned machine learning models and machine-learning-based MPC/EMPC schemes were demonstrated through the applications to chemical process examples.

In Chapter 6, the concept of operational safety in process control was introduced, followed by a novel function termed control Lyapunov-barrier function (CLBF) that was used to derive stability and safety properties. Lyapunov-based MPC and EMPC schemes that incorporate CLBF-based constraints were developed to maintain the process state in the safe operating region and optimize process performance simultaneously. Rigorous theoretical results of closed-loop stability, process operational safety and recursive feasibility of MPCs were developed, and a benchmark chemical reactor example was used to illustrate the effectiveness of the proposed CLBF-based MPC/EMPC methods.

In Chapter 7, issues relating to model development and real-time implementation of CLBF-based MPC schemes were addressed. The CLBF-based MPC and EMPC schemes using RNN models for predicting system dynamics were developed, with sufficient conditions under which closed-loop stability and operational safety were derived. Online learning of machine learning models were implemented within MPCs to update models for the nonlinear process subject to time-varying disturbances. The methods were applied to the benchmark chemical reactor

example.

In Chapter 8, machine-learning-based detection systems and resilient control schemes were developed to detect and mitigate the impact of stealthy cyber-attacks in MPC and EMPC systems. The construction method of data-based machine-learning detectors that can detect multiple classes of intelligent cyber-attacks was first presented. Several cyber-attack resilient control strategies were subsequently developed to contain and eliminate the impact of cyber-attacks by reconfiguring the control system. The application to a benchmark multivariable nonlinear process example was presented to evaluate the ability of the integrated detection and mitigation scheme.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] S. Agrawal and J. Agrawal. Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60:708–713, 2015.

[3] A Ahlén, J Akerberg, M Eriksson, A. J. Isaksson, T Iwaki, K. H. Johansson, S Knorn, T Lindh, and H Sandberg. Toward wireless control in industrial process automation: A case study at a paper mill. *IEEE Control Systems Magazine*, 39:36–57, 2019.

[4] A. Alanqar, H. Durand, and P. D. Christofides. On identification of well-conditioned nonlinear systems: Application to economic model predictive control of nonlinear processes. *AIChE Journal*, 61:3353–3373, 2015.

[5] A. Alanqar, H. Durand, and P. D. Christofides. Error-triggered on-line model identification for model-based feedback control. *AIChE Journal*, 63:949–966, 2017.

[6] A. Alanqar, M. Ellis, and P. D. Christofides. Economic model predictive control of nonlinear process systems using empirical models. *AIChE Journal*, 61:816–830, 2015.

[7] F. Albalawi, H. Durand, and P. D. Christofides. Distributed economic model predictive control for operational safety of nonlinear processes. *AIChE Journal*, 63:3404–3418, 2017.

[8] C. F. Alcala and S. J. Qin. Reconstruction-based contribution for process monitoring with kernel principal component analysis. *Industrial & Engineering Chemistry Research*, 49:7849–7857, 2010.

[9] J. M. Ali, M. A. Hussain, M. O. Tade, and J. Zhang. Artificial intelligence techniques applied as estimator in chemical process systems–a literature survey. *Expert Systems with Applications*, 42:5915–5931, 2015.

[10] G. S. Almasi and A. Gottlieb. Highly parallel computing. 1988.

[11] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. In *Proceedings of the 18th European Control Conference*, pages 3420–3431, Saint Petersburg, Russia, 2019.

[12] A. D. Ames, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 6271–6278, Los Angeles, California, 2014.

[13] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62:3861–3876, 2017.

[14] D. Angeli, R. Amrit, and J. B. Rawlings. On average performance and stability of economic model predictive control. *IEEE Transactions on Automatic Control*, 57:1615–1626, 2012.

[15] G. Ba, Y.and Zhao and A. Kadambi. Blending diverse physical priors with neural networks. *arXiv preprint arXiv:1910.00201*, 2019.

[16] R. F Babiceanu and R. Seker. Big data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook. *Computers in Industry*, 81:128–137, 2016.

[17] E. Bakolas and J. H. Saleh. Augmenting the traditional defense-in-depth strategy with the concept of a diagnosable safety architecture. In R. Briš, C. G. Soares, and S. Martorell, editors, *Reliability, Risk and Safety: Theory and Applications*, volume 3, pages 2113–2122. CRC Press/Balkema, Leiden, Netherlands, 2010.

[18] M. Baldi. Cybersecurity defense for industrial process-control systems. *Chemical Engineering*, 123:36, 2016.

[19] D. A. Beck, J. M. Carothers, V. R. Subramanian, and J. Pfaendtner. Data science: Accelerating innovation and discovery in chemical engineering. *AIChE Journal*, 62:1402–1416, 2016.

[20] S. A. Billings. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

[21] C. M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7:108–116, 1995.

[22] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[23] P. Braun and C. M. Kellett. On (the existence of) control Lyapunov barrier functions. *Preprint, https://eref. uni-bayreuth. de/40899*, 2018.

[24] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18:1153–1176, 2015.

[25] F. Burden and D. Winkler. Bayesian regularization of neural networks. In *Artificial Neural Networks*, pages 23–42. Springer, New York, NY, 2008.

[26] E. F. Camacho and C. B. Alba. *Model Predictive Control*. Springer, 2nd edition, 2013.

[27] A. A. Cárdenas, S. Amin, Z. S. Lin, Y. L. Huang, C. Y. Huang, and S. Sastry. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM symposium on information, computer and communications security*, pages 355–366. ACM, 2011.

[28] S. Chen, Z. Wu, and P. D. Christofides. Cyber-attack detection and resilient operation of nonlinear processes under economic model predictive control. *Computers & Chemical Engineering*, 136:106806, 2020.

[29] S. Chen, Z. Wu, and P. D. Christofides. A cyber-secure control-detector architecture for nonlinear processes. *AIChE Journal*, 66:e16907, 2020.

[30] S. Chen, Z. Wu, and P. D. Christofides. Decentralized machine-learning-based predictive control of nonlinear processes. *Chemical Engineering Research and Design*, 162:45–60, 2020.

[31] S. Chen, Z. Wu, D. Rincon, and P. D. Christofides. Machine learning-based distributed model predictive control of nonlinear processes. *AIChE Journal*, 66:e17013, 2020.

[32] Y. Chen, H. Peng, and J. Grizzle. Obstacle avoidance for low-speed autonomous vehicles with barrier function. *IEEE Transactions on Control Systems Technology*, 26:194–206, 2017.

[33] F. Chollet et al. Keras. https://keras.io, 2015.

[34] Tommy WS Chow and Yong Fang. A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics. *IEEE Transactions on Industrial Electronics*, 45:151–161, 1998.

[35] P. D. Christofides, J. F. Davis, N. H. El-Farra, D. Clark, K. R. Harris, and J. N. Gipson. Smart plant operations: Vision, progress and challenges. *AIChE Journal*, 53:2734–2741, 2007.

[36] P. D. Christofides, J. Liu, and D. Muñoz de la Peña. *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications*. Advances in Industrial Control Series. Springer-Verlag, London, England, 2011.

[37] R. V. Cowlagi and J. H. Saleh. Coordinability and consistency: Application of systems theory to accident causation and prevention. *Journal of Loss Prevention in the Process Industries*, 33:200–212, 2015.

[38] A. Cozad, N. V. Sahinidis, and D. C. Miller. A combined first-principles and data-driven approach to model building. *Computers & Chemical Engineering*, 73:116–127, 2015.

[39] T. Cruz, L. Rosa, J. Proença, L. Maglaras, M. Aubigny, L. Lev, J. Jiang, and P. Simoes. A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Transactions on Industrial Informatics*, 12:2236–2246, 2016.

[40] L. D. Dalcin, R. R. Paz, P. A. Kler, and A. Cosimo. Parallel distributed computing using python. *Advances in Water Resources*, 34:1124–1139, 2011.

[41] S.M. Dibaji, M. Pirani, D.B. Flamholz, A.M. Annaswamy, K.H. Johansson, and A. Chakrabortty. A systems and control perspective of CPS security. *Annual Reviews in Control*, 2019.

[42] D. Ding, Q. L. Han, Y. Xiang, X. Ge, and X. M. Zhang. A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 275:1674–1683, 2018.

[43] M. Ellis, H. Durand, and P. D. Christofides. A tutorial review of economic model predictive control methods. *Journal of Process Control*, 24:1156–1178, 2014.

[44] H. Fawzi, P. Tabuada, and S. Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, 59:1454–1467, 2014.

[45] S. Feyo de Azevedo, B. Dahm, and F.R. Oliveira. Hybrid modelling of biochemical processes: A comparison with the conventional approach. *Computers & Chemical Engineering*, 21:S751–S756, 1997.

[46] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice–A survey. *Automatica*, 25:335–348, 1989.

[47] H. W. Ge, Y. C. Liang, and M. Marchese. A modified particle swarm optimization-based dynamic recurrent neural network for identifying and controlling nonlinear systems. *Computers & Structures*, 85:1611–1622, 2007.

[48] S. S. Ge and C. Wang. Adaptive neural control of uncertain mimo nonlinear systems. *IEEE Transactions on Neural Networks*, 15(3):674–692, 2004.

[49] M. Gentile, W. J. Rogers, and M. S. Mannan. Development of an inherent safety index based on fuzzy logic. *AIChE Journal*, 49:959–968, 2003.

[50] P. Georgieva, M. J. Meireles, and S. F. de Azevedo. Knowledge-based hybrid modelling of a batch crystallisation when accounting for nucleation, growth and agglomeration phenomena. *Chemical Engineering Science*, 58:3699–3713, 2003.

[51] J. Gong and F. You. Optimal design and synthesis of algal biorefinery processes for biological carbon sequestration and utilization with zero direct greenhouse gas emissions: MINLP model and global optimization algorithm. *Industrial & Engineering Chemistry Research*, 53:1563–1579, 2014.

[52] J. P. Gupta and D. W. Edwards. Inherently safer design - Present and future. *Process Safety and Environmental Protection*, 80:115–125, 2002.

[53] K. Gurney. *An introduction to neural networks*. CRC press, 2014.

[54] M. F Harkat, S. Djelel, N. Doghmane, and M. Benouaret. Sensor fault detection, isolation and reconstruction using nonlinear principal component analysis. *International Journal of Automation and Computing*, 4:149–155, 2007.

[55] C. He and F. You. Shale gas processing integrated with ethylene production: Novel process designs, exergy analysis, and techno-economic analysis. *Industrial & Engineering Chemistry Research*, 53:11442–11459, 2014.

[56] W. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 3270–3285, Maui, Hawaii, 2012.

[57] M. Heidarinejad, J. Liu, and P. D. Christofides. Economic model predictive control of nonlinear process systems using Lyapunov techniques. *AIChE Journal*, 58:855–870, 2012.

[58] A-M. Heikkilä, M. Hurme, and M. Järveläinen. Safety considerations in process synthesis. *Computers & Chemical Engineering*, 20:S115–S120, 1996.

[59] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan. Machine learning for power system disturbance and cyber-attack discrimination. In *Proceedings of the 7th International Symposium on Resilient Control Systems*, pages 1–8, Denver, CO, USA, 2014. IEEE.

[60] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: a tutorial. *Statistical Science*, pages 382–401, 1999.

[61] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[62] L. Huang, X. Nguyen, M. N. Garofalakis, J. M. Hellerstein, M. I. Jordan, A. D. Joseph, and N. Taft. Communication-efficient online detection of network-wide anomalies. In *INFOCOM*, volume 7, pages 134–142, Anchorage, Alaska, USA, 2007.

[63] R. Huang, L. T. Biegler, and E. Harinath. Robust stability of economically oriented infinite horizon NMPC that include cyclic processes. *Journal of Process Control*, 22:51–59, 2012.

[64] P. Jain, H. J. Pasman, S. Waldram, E. N. Pistikopoulos, and M. S. Mannan. Process resilience analysis framework (PRAF): A systems approach for improved risk and safety management. *Journal of Loss Prevention in the Process Industries*, 53:61–73, 2018.

[65] M. Jankovic. Combining control Lyapunov and barrier functions for constrained stabilization of nonlinear systems. In *Proceedings of the American Control Conference*, pages 1916–1922, Seattle, Washington, 2017.

[66] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[67] K. N. Junejo and J. Goh. Behaviour-based attack detection and classification in cyber physical systems using machine learning. In *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, pages 34–43, 2016.

[68] O. Kahrs and W. Marquardt. The validity domain of hybrid models and its application in process optimization. *Chemical Engineering and Processing: Process Intensification*, 46:1054–1066, 2007.

[69] O. Kahrs and W. Marquardt. Incremental identification of hybrid process models. *Computers & Chemical Engineering*, 32:694–705, 2008.

[70] A. Karpatne, W. Watkins, J. Read, and V. Kumar. Physics-guided neural networks (PGNN): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

[71] M. Kellman, E. Bostan, N. Repina, and L. Waller. Physics-based learned design: Optimized coded-illumination for quantitative phase imaging. *arXiv preprint arXiv:1808.03571*, 2019.

[72] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ, third edition, 2002.

[73] F. I. Khan and P. R. Amyotte. How to make inherent safety practice a reality. *The Canadian Journal of Chemical Engineering*, 81:2–16, 2003.

[74] M. Kheradmandi and P. Mhaskar. Data driven economic model predictive control. *Mathematics*, 6:51, 2018.

[75] F. Khorrami, P. Krishnamurthy, and R. Karri. Cybersecurity for control systems: A process-aware perspective. *IEEE Design & Test*, 33:75–83, 2016.

[76] K. Kidam and M. Hurme. Analysis of equipment failures as contributors to chemical process accidents. *Process Safety and Environmental Protection*, 91:61–78, 2013.

[77] M. Kim, H. Liu, J. T. Kim, and C. Yoo. Sensor fault identification and reconstruction of indoor air quality (IAQ) data using a multivariate non-gaussian model in underground building space. *Energy and Buildings*, 66:384–394, 2013.

[78] T. Kletz. *What Went Wrong? - Case Histories of Process Plant Disasters and How They Could Have Been Avoided*. Elsevier, Burlington, Massachusetts, fifth edition, 2009.

[79] W. Knowles, D. Prince, D. Hutchison, J.F.P Disso, and K. Jones. A survey of cyber security management in industrial control systems. *International Journal of Critical Infrastructure Protection*, 9:52–80, 2015.

[80] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou. High-order neural network structures for identification of dynamical systems. *IEEE Transactions on Neural Networks*, 6:422–431, 1995.

[81] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2:164–168, 1944.

[82] N. G. Leveson and G. Stephanopoulos. A system-theoretic, control-inspired view and approach to process safety. *AIChE Journal*, 60:2–14, 2014.

[83] Y. Lin and E. D. Sontag. A universal formula for stabilization with bounded controls. *Systems & Control Letters*, 16:393–397, 1991.

[84] D. Liu and G. H. Yang. Neural network-based event-triggered mfac for nonlinear discrete-time processes. *Neurocomputing*, 272:356–364, 2018.

[85] P. Liu, E. N. Pistikopoulos, and Z. Li. A multi-objective optimization approach to polygeneration energy systems design. *AIChE Journal*, 56:1218–1234, 2010.

[86] Z. Long, Y. Lu, X. Ma, and B. Dong. PDE-net: Learning PDEs from data. *arXiv preprint arXiv:1710.09668*, 2017.

[87] Y. Lu, M. Rajora, P. Zou, and S. Liang. Physics-embedded machine learning: case study with electrochemical micro-machining. *Machines*, 5:4–15, 2017.

[88] M. L. Luyben, B. D. Tyreus, and W. L. Luyben. Plantwide control design procedure. *AIChE Journal*, 43:3161–3174, 1997.

[89] M.S. Mannan, S. Sachdeva, H. Chen, O. Reyes-Valdes, Y. Liu, and D.M. Laboureur. Trends and challenges in process safety. *AIChE Journal*, 61:3558, 2015.

[90] S. Mannan. *Lees' Loss Prevention in the Process Industries - Hazard Identification, Assessment and Control*. Elsevier, Waltham, Massachusetts, fourth edition, 2012.

[91] T. Marlin. *Operability in Process Design: Achieving Safe, Profitable, and Robust Process Operations*. McMaster University, Ontario, Canada, 2012.

[92] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11:431–441, 1963.

[93] W. Marquardt. Nonlinear model reduction for optimization based control of transient chemical processes. In *AIChE Symposium Series*, pages 12–42. New York; American Institute of Chemical Engineers; 1998, 2002.

[94] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[95] A. Mehra, W. Ma, F. Berg, P. Tabuada, J. W. Grizzle, and A. D. Ames. Adaptive cruise control: Experimental validation of advanced controllers on scale-model cars. In *Proceedings of the American Control Conference*, pages 1411–1418, Chicago, Illinois, 2015.

[96] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa. Ensemble approaches for regression: A survey. *ACM Computing Surveys*, 45:10, 2012.

[97] D. I. Mendoza-Serrano and D. J. Chmielewski. Smart grid coordination in building HVAC systems: EMPC and the impact of forecasting. *Journal of Process Control*, 24:1301–1310, 2014.

[98] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Transactions on Automatic Control*, 50:1670–1680, 2005.

[99] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Systems & Control Letters*, 55:650–659, 2006.

[100] S. R. Mohanty, A. K. Pradhan, and A. Routray. A cumulative sum-based fault detector for power system relaying application. *IEEE Transactions on Power Delivery*, 23:79–86, 2007.

[101] M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23:667–682, 1999.

[102] Manfred Morari and Jay H Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23:667–682, 1999.

[103] D. Muñoz de la Peña and P. D. Christofides. Lyapunov-based model predictive control of nonlinear systems subject to data losses. *IEEE Transactions on Automatic Control*, 53:2076–2089, 2008.

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

[105] Y. L. Murphey, M. A. Masrur, Z. H. Chen, and B. Zhang. Model-based fault diagnosis in electric drives using machine learning. *IEEE/ASME Transactions on Mechatronics*, 11:290–303, 2006.

[106] K. S. Narendra and A. M. Annaswamy. *Stable Adaptive Systems*. Courier Corporation, 2012.

[107] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1:4–27, 1990.

[108] W. Ni, S. K. Tan, W. J. Ng, and S. D. Brown. Moving-window gpr for nonlinear dynamic system modeling with dual updating and dual preprocessing. *Industrial & Engineering Chemistry Research*, 51:6416–6428, 2012.

[109] B. Niu and J. Zhao. Barrier Lyapunov functions for the output tracking control of constrained nonlinear switched systems. *Systems & Control Letters*, 62:963–971, 2013.

[110] A. M. Niziolek, O. Onel, M. M. F. Hasan, and C. A. Floudas. Municipal solid waste to liquid transportation fuels - Part II: Process synthesis and global optimization strategies. *Computers & Chemical Engineering*, 74:184–203, 2015.

[111] R. M. Noor, Z. Ahmad, M. M. Don, and M. H. Uzir. Modelling and control of different types of polymerization processes using neural networks technique: a review. *The Canadian Journal of Chemical Engineering*, 88:1065–1084, 2010.

[112] S. Omar, A. Ngadi, and H. H. Jebur. Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, 79:33–41, 2013.

[113] B. P. Omell and D. J. Chmielewski. IGCC power plant dispatch using infinite-horizon economic model predictive control. *Industrial & Engineering Chemistry Research*, 52:3151–3164, 2013.

[114] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor. Machine learning methods for attack detection in the smart grid. *IEEE Transactions on Neural Networks and Learning Systems*, 27:1773–1786, 2015.

[115] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon. Identification of nonlinear systems using polynomial nonlinear state space models. *Automatica*, 46:647–656, 2010.

[116] Y. Pan and J. Wang. Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks. *IEEE Transactions on Industrial Electronics*, 59:3089–3101, 2011.

[117] B. A. Pearlmutter. Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural Networks*, 6:1212–1228, 1995.

[118] C. Peng, H. Sun, M. Yang, and Y. Wang. A survey on security communication and control for smart grids under malicious cyber attacks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[119] M. M. Polycarpou and P. A. Ioannou. Identification and control of nonlinear systems using neural network models: Design and stability analysis. Technical report, University of Southern California, 1991.

[120] C.R. Porfirio, E. A. Neto, and D. Odloak. Multi-model predictive control of an industrial C3/C4 splitter. *Control Engineering Practice*, 11:765–779, 2003.

[121] S. Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42:117–126, 2006.

[122] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Proceedings of the 7th International Workshop, HSCC*, volume 2993, pages 477–492, Philadelphia, Pennsylvania, 2004.

[123] D. C. Psichogios and L. H. Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38:1499–1511, 1992.

[124] S. J. Qin. Survey on data-driven industrial process monitoring and diagnosis. *Annual reviews in control*, 36:220–234, 2012.

[125] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[126] S. J. Qin and L. H. Chiang. Advances and opportunities in machine learning for process data analytics. *Computers & Chemical Engineering*, 126:465–473, 2019.

[127] S. J. Qin and R. Dunia. Determining the number of principal components for best reconstruction. *Journal of Process Control*, 10:245–250, 2000.

[128] J Raiyn. A survey of cyber attack detection strategies. *International Journal of Security and its Applications*, 8:247–256, 2014.

[129] J. B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 20:38–52, 2000.

[130] J. B. Rawlings and C. T. Maravelias. Bringing new technologies and approaches to the operation and control of chemical process systems. *AIChE Journal*, 65:e16615.

[131] S. Rhode, S. Hong, J. K. Hedrick, and F. Gauterin. Vehicle tractive force prediction with robust and windup-stable kalman filters. *Control Engineering Practice*, 46:37–50, 2016.

[132] M. Rodrigues, D. Theilliol, M. Adam-Medina, and D. Sauter. A fault detection and isolation scheme for industrial systems based on multiple operating models. *Control Engineering Practice*, 16:225–239, 2008.

[133] M. Z. Romdlony and B. Jayawardhana. Stabilization with guaranteed safety using control Lyapunov–barrier function. *Automatica*, 66:39–47, 2016.

[134] A. Sahoo, H. Xu, and S. Jagannathan. Neural network-based event-triggered state feedback control of nonlinear continuous-time systems. *IEEE Transactions on Neural Networks and Learning Systems*, 27:497–509, 2015.

[135] B. Samanta, K. R. Al-Balushi, and S. A. Al-Araimi. Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. *Engineering Applications of Artificial Intelligence*, 16:657–665, 2003.

[136] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[137] Q. Shen, B. Jiang, P. Shi, and C. C. Lim. Novel neural networks-based fault tolerant control scheme with fault alarm. *IEEE Transactions on Cybernetics*, 44:2190–2201, 2014.

[138] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S. Chung. Neural lander: Stable drone landing control using learned dynamics. In *Proceedings of the International Conference on Robotics and Automation*, pages 9784–9790, Montreal, Canada, 2019.

[139] P. Sibi, S. A. Jones, and P. Siddarth. Analysis of different activation functions using back propagation neural networks. *Journal of Theoretical and Applied Information Technology*, 47:1264–1268, 2013.

[140] J. Singh and M. J. Nene. A survey on machine learning techniques for intrusion detection systems. *International Journal of Advanced Research in Computer and Communication Engineering*, 2:4349–4355, 2013.

[141] E. D. Sontag. Neural nets as systems models and controllers. In *Proceedings of the Seventh Yale Workshop on Adaptive and Learning Systems*, pages 73–79, Yale University, 1992.

[142] E. D. Sontag. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, 13:117–123, 1989.

[143] G. Stephanopoulos and C. Han. Intelligent systems in process engineering: A review. *Computers & Chemical Engineering*, 20:743–791, 1996.

[144] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52:1680–1685, 2007.

[145] K. P. Tee, S. S. Ge, and E. H. Tay. Barrier Lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45:918–927, 2009.

[146] M. L. Thompson and M. A. Kramer. Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal*, 40:1328–1340, 1994.

[147] Y. Tian, J. Zhang, and J. Morris. Modeling and optimal control of a batch polymerization reactor using a hybrid stacked recurrent neural network model. *Industrial & Engineering Chemistry Research*, 40:4525–4535, 2001.

[148] A. P. Trischler and G. M. D'Eleuterio. Synthesis of recurrent neural networks for dynamical system simulation. *Neural Networks*, 80:67–78, 2016.

[149] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36:11994–12000, 2009.

[150] R. Turton, R. C. Bailie, W. B. Whiting, and J. A. Shaeiwitz. *Analysis, synthesis and design of chemical processes*. Pearson Education, 2008.

[151] J. Valappil and C. Georgakis. Systematic estimation of state noise statistics for extended kalman filters. *AIChE Journal*, 46:292–308, 2000.

[152] A. Van Mulders, J. Schoukens, M. Volckaert, and M. Diehl. Two nonlinear optimization methods for black box identification compared. *Automatica*, 46:1675–1681, 2010.

[153] P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93, 1994.

[154] V. Venkatasubramanian. Systemic failures: Challenges and opportunities in risk management in complex systems. *AIChE Journal*, 57:2–9, 2011.

[155] V. Venkatasubramanian. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, 65:466–478, 2019.

[156] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin. A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers & Chemical Engineering*, 27:327–346, 2003.

[157] M. Viberg. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31:1835–1851, 1995.

[158] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[159] L. Wang, A. D. Ames, and M. Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33:661–674, 2017.

[160] S. Wang and Y. Chen. Sensor validation and reconstruction for building central chilling systems based on principal component analysis. *Energy Conversion and Management*, 45:673–695, 2004.

[161] X. Wang and M. D. Lemmon. Event design in event-triggered feedback control systems. *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 2105–2110, Cancun, Mexico, 2008.

[162] Y. Wang. A new concept using lstm neural networks for dynamic system identification. In *Proceedings of the American Control Conference*, pages 5324–5329, Seattle, Washington, 2017.

[163] S. R. West, Y. Guo, X. R. Wang, and J. Wall. Automated fault detection and diagnosis of HVAC subsystems using statistical machine learning. In *Proceedings of the 12th International Conference of the International Building Performance Simulation Association*, Sydney, Australia, 2011.

[164] A. Widodo and B. Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21:2560–2574, 2007.

[165] P. Wieland and F. Allgöwer. Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40:462–467, 2007.

[166] Z. T. Wilson and N. V. Sahinidis. The alamo approach to machine learning. *Computers & Chemical Engineering*, 106:785–795, 2017.

[167] W. Wong, E. Chee, J. Li, and X. Wang. Recurrent neural network-based model predictive control for continuous pharmaceutical manufacturing. *Mathematics*, 6:242, 2018.

[168] Z. Wu, F. Albalawi, J. Zhang, Z. Zhang, H. Durand, and P. D. Christofides. Detecting and handling cyber-attacks in model predictive control of chemical processes. *Mathematics*, 6(10):173, 2018.

[169] Z. Wu, F. Albalawi, Z. Zhang, J. Zhang, H. Durand, and P. D. Christofides. Control Lyapunov-barrier function-based model predictive control of nonlinear systems. *Automatica*, 109:108508, 2019.

[170] Z. Wu, S. Chen, D. Rincon, and P. D. Christofides. Post cyber-attack state reconstruction for nonlinear processes using machine learning. *Chemical Engineering Research and Design*, 159:248–261, 2020.

[171] Z. Wu and P. D. Christofides. Economic machine-learning-based predictive control of nonlinear systems. *Mathematics*, 7(6):494, 20 pages, 2019.

[172] Z. Wu and P. D. Christofides. Handling bounded and unbounded unsafe sets in control Lyapunov-barrier function-based model predictive control of nonlinear processes. *Chemical Engineering Research and Design*, 143:140–149, 2019.

[173] Z. Wu and P. D. Christofides. Optimizing process economics and operational safety via economic MPC using barrier functions and recurrent neural network models. *Chemical Engineering Research and Design*, 152:455–465, 2019.

[174] Z. Wu and P. D. Christofides. Control Lyapunov-barrier function-based predictive control of nonlinear processes using machine learning modeling. *Computers & Chemical Engineering*, 134:106706, 2020.

[175] Z. Wu, H. Durand, and P. D. Christofides. Safe economic model predictive control of nonlinear systems. *Systems & Control Letters*, 118:69–76, 2018.

[176] Z. Wu, Q. S. Jia, and X. Guan. Optimal control of multiroom hvac system: An event-based approach. *IEEE Transactions on Control Systems Technology*, 24(2):662–669, 2015.

[177] 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*, 89:74–84, 2020.

[178] Z. Wu, D. Rincon, and P. D. Christofides. Real-time adaptive machine-learning-based predictive control of nonlinear processes. *Industrial & Engineering Chemistry Research*, 59:2275–2290, 2020.

[179] Z. Wu, D. Rincon, and P. D. Christofides. Real-time machine learning for operational safety of nonlinear processes via barrier-function based predictive control. *Chemical Engineering Research and Design*, 155:88–97, 2020.

[180] Z. Wu, A. Tran, Y. M. Ren, C. S. Barnes, S. Chen, and P. D. Christofides. Model predictive control of phthalic anhydride synthesis in a fixed-bed catalytic reactor via machine learning modeling. *Chemical Engineering Research and Design*, 145:173–183, 2019.

[181] Z. Wu, A. Tran, D. Rincon, and P. D. Christofides. Machine learning-based predictive control of nonlinear processes. part I: Theory. *AIChE Journal*, 65:e16729, 2019.

[182] Z. Wu, A. Tran, D. Rincon, and P. D. Christofides. Machine learning-based predictive control of nonlinear processes. part II: Computational implementation. *AIChE Journal*, 65:e16734, 2019.

[183] Q. Xiong and A. Jutan. Grey-box modelling and control of chemical processes. *Chemical Engineering Science*, 57:1027–1039, 2002.

[184] J. Xu, C. Li, X. He, and T. Huang. Recurrent neural network for solving model predictive control problem in application of four-tank benchmark. *Neurocomputing*, 190:172–178, 2016.

[185] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*, 48:54–61, 2015.

[186] N. Ye, Y. Zhang, and C. M. Borror. Robustness of the markov-chain model for cyber-attack detection. *IEEE Transactions on Reliability*, 53:116–123, 2004.

[187] S. Yin and O. Kaynak. Big data for modern industry: challenges and trends [point of view]. *Proceedings of the IEEE*, 103:143–146, 2015.

[188] V. M. Zavala. A multiobjective optimization perspective on the stability of economic MPC. In *Proceedings of the 9th IFAC Symposium on Advanced Control of Chemical Processes*, pages 975–981, Whistler, Canada, 2015.

[189] C. Zhang and Y. Ma. *Ensemble machine learning: methods and applications*. Springer, 2012.

[190] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler. Machine learning and deep learning algorithms for bearing fault diagnostics-a comprehensive review. *arXiv preprint arXiv:1901.08247*, 2019.

[191] Z. Zhang, Z. Wu, H. Durand, F. Albalawi, and P. D. Christofides. On integration of feedback control and safety systems: Analyzing two chemical process applications. *Chemical Engineering Research and Design*, 132:616–626, 2018.

[192] Z. Zhang, Z. Wu, D. Rincon, and P. D. Christofides. Real-time optimization and control of nonlinear processes using machine learning. *Mathematics*, 7:890, 2019.

[193] L.F.M. Zorzetto, R. Maciel Filho, and M.R. Wolf-Maciel. Processing modelling development through artificial neural networks and hybrid models. *Computers & Chemical Engineering*, 24:1355–1360, 2000.