Title

Integrated Encrypted Model Predictive Control Systems for Cyber-Resilient Operation of Nonlinear Processes

Author

Kadakia, Yash

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Integrated Encrypted Model Predictive Control Systems for Cyber-Resilient Operation of Nonlinear Processes

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Science
in Chemical Engineering

by

Yash Ashit Kadakia

2024

ABSTRACT OF THE THESIS

Integrated Encrypted Model Predictive Control Systems for Cyber-Resilient Operation of Nonlinear Processes

by

Yash Ashit Kadakia

Master of Science in Chemical Engineering

University of California, Los Angeles, 2024

Professor Panagiotis D. Christofides, Chair

In industrial environments, the collection of vast amounts of operational and instrumentation data serves critical purposes such as monitoring, control, preventative maintenance, fault detection, and troubleshooting. Networked control systems have revolutionized traditional methodologies, offering seamless data transfer capabilities while minimizing wiring and maintenance issues. Their ease of implementation and scalability make them applicable across a wide spectrum of operations, from small-scale setups to large industrial complexes. However, the efficient functioning of industrial process control systems in real-time heavily relies on the accuracy of recorded data and the dependability of networked communication channels. Any compromise in the integrity or con-

fidentiality of this data due to unauthorized access or manipulation by malicious entities can result in severe consequences, impacting operational safety and economic performance. As intelligent cyber-attacks have the potential to access system information, it is necessary to develop networked control systems that maintain the confidentiality of industrial data, and have cyber-attack detection and resilient operation strategies to address cybersecurity issues beyond fault diagnosis, and is the focus of this thesis.

Large-scale industrial processes encounter numerous control and operational challenges, such as nonlinearity, high dimensionality, complex interacting process dynamics, inherent state and input delays, and limited sensor measurements. To address these challenges effectively, a comprehensive mathematical model representing plant dynamics is essential, with appropriate integrations to tackle specific challenges. For example, model predictive control systems can handle multivariable interactions and input/state constraints, state predictors can address input delays, time-lag models are employed to account for state delays, and observers are integrated to estimate unavailable data accurately. Additionally, distributed and decentralized control structures offer improved computational efficiency compared to centralized frameworks, particularly advantageous for large-scale processes. Real-time adaptation to fluctuating economics is another crucial aspect for maintaining competitiveness in the market. Balancing these challenges with the need to enhance cybersecurity and ensure the confidentiality of system data necessitates the development of innovative control frameworks.

Motivated by the above, this thesis introduces novel control architectures featuring encrypted communication tailored for various nonlinear processes. Encryption techniques are integrated into centralized, decentralized, and distributed model predictive control (MPC) systems. Additionally,

this thesis introduces two-layer and two-tier encrypted control frameworks that combine linear and nonlinear control strategies. Within these frameworks, linear control systems perform control input computations in an encrypted space, eliminating the need for decryption and ensuring the preservation of data confidentiality. Incorporating machine-learning-based and logic-based cyberattack detectors with reconfiguration mechanisms further fortifies these encrypted control frameworks for cyber-resilient operation. System-specific integrations in the control system address complexities like limited feedback, input and state delays, and the dynamic nature of process economics. Numerical simulations of nonlinear chemical process examples and Aspen Plus simulations of large-scale chemical process networks demonstrate the effectiveness of the proposed frameworks. The results highlight their ability to improve operational safety, cyber-security, computational efficiency, and overall closed-loop and economic performance in nonlinear processes.

The thesis of Yash Ashit Kadakia is approved.

Dante A. Simonetti

Carlos G. Morales Guio

Panagiotis D. Christofides, Committee Chair

University of California, Los Angeles

2024

Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Professor Panagiotis D. Christofides, for his support, encouragement, and guidance on both my technical work and professional development during my research. Professor Christofides sets an example of excellence as a researcher, mentor, instructor, and role model. I am lucky to be a student of his, as this M.S. experience has laid out a strong foundation for my future career.

I would like to thank Professors Dante Simonetti and Carlos Morales-Guio for reviewing my thesis and contributing to my Master's thesis committee.

In addition, I would like to thank all of my colleagues in the Christofides research group, including Henrik Wang, Feiyang Ou, Xiaodong Cui, Dominic Peters, Parth Chheda, Yifei Wang, Esther Hsu, Arthur Khodaverdian, Julius Suherman, Dhruv Gohil, and Berkay Citmaci. I would like to particularly thank Dr. Fahim Abdullah, Atharva Suryavanshi, Aisha Alnajdi, and Matthew Tom, who worked closely with me to tackle significant challenges in my M.S. journey.

Last but not least, I am indebted to my parents, Alpa Kadakia and Ashit Kadakia. They have always believed in me, and it is their unconditional love, unwavering support and affirmation that gave me the strength and motivation to overcome the next challenge in life.

With regard to the research that forms the foundation for this work:

Chapter 2 contains versions of: Kadakia, Y. A., A. Suryavanshi, A. Alnajdi, F. Abdullah, and P. D. Christofides, "Encrypted Model Predictive Control of a Nonlinear Chemical Process Network," *Processes*, 11 (8), 2501, 2023.

Chapter 3 contains versions of: Kadakia, Y. A., A. Suryavanshi, A. Alnajdi, F. Abdullah, and

P. D. Christofides, "Integrating Machine Learning Detection and Encrypted Control for Enhanced Cybersecurity of Nonlinear Processes,"*Comp. & Chem. Eng.*, 180, 108498, 2024.

Chapter 4 contains versions of: Kadakia, Y. A., A. Alnajdi, F. Abdullah, and P. D. Christofides, "Encrypted Decentralized Model Predictive Control of Nonlinear Processes with Delays," *Chem. Eng. Res. & Des.*, 200, 312–324, 2023.

Chapter 5.1 and Chapter 5.2 contain versions of: Kadakia, Y. A., A. Alnajdi, F. Abdullah, and P. D. Christofides, "Encrypted Distributed Model Predictive Control with State Estimation for Nonlinear Processes," *Dig. Chem. Eng.*, 9, 100133, 2023; Kadakia, Y. A., F. Abdullah, A. Alnajdi, and P. D. Christofides, "Encrypted distributed model predictive control of nonlinear processes," *Contr. Eng. Pract.*, 145, 105874, 2024.

Chapter 6 contains versions of: Kadakia, Y. A., F. Abdullah, A. Alnajdi, and P. D. Christofides, "Integrating dynamic economic optimization and encrypted control for cyber-resilient operation of nonlinear processes," *AIChE J.*, Submitted, 2024.

Curriculum Vitae

Education

University of Mumbai Aug 2018 - May 2022

B.E., Chemical Engineering **Mumbai, India**

Journal Publications

1. **Kadakia, Y. A.**, F. Abdullah, A. Alnajdi, and P. D. Christofides, "Integrating dynamic economic optimization and encrypted control for cyber-resilient operation of nonlinear processes," *AIChE J.*, Submitted, 2024.

2. **Kadakia, Y. A.**, F. Abdullah, A. Alnajdi, and P. D. Christofides, "Encrypted distributed model predictive control of nonlinear processes," *Contr. Eng. Pract.*, **145**, 105874, 2024.

3. **Kadakia, Y. A.**, A. Alnajdi, F. Abdullah, and P. D. Christofides, "Encrypted Distributed Model Predictive Control with State Estimation for Nonlinear Processes," *Dig. Chem. Eng.*, **9**, 100133, 2023.

4. **Kadakia, Y. A.**, A. Alnajdi, F. Abdullah, and P. D. Christofides, "Encrypted Decentralized Model Predictive Control of Nonlinear Processes with Delays," *Chem. Eng. Res. & Des.*, **200**, 312–324, 2023.

5. **Kadakia, Y. A.**, A. Suryavanshi, A. Alnajdi, F. Abdullah, and P. D. Christofides, "Integrating Machine Learning Detection and Encrypted Control for Enhanced Cybersecurity of Nonlinear Processes,"*Comp. & Chem. Eng.*, **180**, 108498, 2024.

6. **Kadakia, Y. A.**, A. Suryavanshi, A. Alnajdi, F. Abdullah, and P. D. Christofides, "Encrypted Model Predictive Control of a Nonlinear Chemical Process Network," *Processes*, **11 (8)**, 2501, 2023.

Presentations

1. **Kadakia, Y. A.**, A. Suryavanshi, A. Alnajdi, F. Abdullah and P. D. Christofides, "A Two-Tier Encrypted Control Architecture for Enhanced Cybersecurity of Nonlinear Processes," *Proceedings of the American Control Conference*, in press, Toronto, Canada, 2024.

2. **Kadakia, Y. A.**, A. Alnajdi, F. Abdullah and P. D. Christofides, "Encrypted Decentralized Model Predictive Control of Nonlinear Processes with Input Delays," *Proceedings of the American Control Conference*, in press, Toronto, Canada, 2024.

Chapter 1

Introduction

1.1 Motivation

The advent of Industry 4.0 is characterized by widespread sensor deployment, expanded wireless communication capabilities, and more accessible computing power, marking a significant leap in industrial technology. Over the past decade, there has been an exponential growth in data collection and computing capabilities, enabling the development of advanced analytics and intelligent systems [22, 42, 73]. Production facilities now accumulate vast amounts of operational and instrumentation data crucial for monitoring, control, and troubleshooting purposes. Some potential troubleshooting applications of this data include maintenance, fault detection, and building of data-based models, particularly with the computational power available [72, 88]. This digital revolution opens doors to explore more robust systems aimed at enhancing operational stability, confidentiality of data collected, process safety, production quality, computational speed, economic perfor-

mance, and cybersecurity across various industrial domains. In the subsequent sections, we delve into some key challenges within the realm of process control that serve as the driving force behind the research endeavors presented in this thesis.

Industrial data serves as a cornerstone for numerous applications across diverse industries. However, safeguarding confidentiality and ensuring secure access to this data are of utmost importance, particularly in highly competitive markets. The transmission of raw data exposes it to vulnerabilities, including unauthorized access and manipulation by external parties. To counteract these risks, robust measures must be implemented that uphold confidentiality and restrict access to authorized personnel, even during data transmission over networks. Moreover, the approach adopted should be universally applicable across different industries, minimizing the need for industry-specific measures to ensure data confidentiality. A promising solution is utilizing an encrypted control system [30, 46]. This approach offers a versatile and effective solution for enhancing data security and confidentiality. It can be easily implemented across various systems without necessitating system-specific modifications, thereby addressing the fundamental challenge of secure data transmission in networked systems.

To implement encryption, most cryptosystems require data to be encrypted in the form of positive integers. However, sensor data is typically in floating point format. Converting this floating point data to positive integers necessitates a mapping procedure. This procedure involves quantization, where a quantization parameter is selected to perform the necessary operations [19]. However, this mapping process introduces errors between the actual value and its encrypted counterpart due to the discrepancy between the real number and its closest rational approximation during mapping. Consequently, to mitigate this issue, the control system must be designed to accommodate a

certain level of robustness concerning potential encryption process errors. Additionally, the mapping procedure should be robust to minimize errors by appropriately adjusting quantization-related parameters.

Various encryption methods, including symmetric encryption, fully homomorphic encryption, and partially homomorphic encryption, serve to secure data. Symmetric encryption, like AES (Advanced Encryption Standard), is a non-homomorphic technique that prohibits mathematical operations within encrypted data [70]. In contrast, fully homomorphic encryption, as seen in schemes like BGV (Brakerski-Gentry-Vaikuntanathan), allows addition and multiplication operations within encrypted data [32]. Meanwhile, partially homomorphic encryption enables addition or multiplication operations within encrypted data. For example, the Paillier cryptosystem supports addition operations in an encrypted environment [67]. However, a key limitation of encryption methods is their inability to perform nonlinear mathematical operations within encrypted data, restricting them to linear additive and multiplicative operations. Nonlinear computations necessitate decryption before processing, underscoring the importance of a cyber-secure decryption environment. The vulnerability of the decryption environment to cyber threats poses challenges for ensuring cyber-resilient operation. Special control structures with multi-tier control are required to address these challenges effectively.

Many industrial applications encounter significant control challenges due to the scale and intricacy of their processes. Conventional methods for analyzing and controlling dynamical systems often rely on the assumption of centrality, wherein a single controller processes all available system information for relevant calculations. However, numerous industrial systems, including chemical production plants, power distribution grids, urban traffic networks, and cyber-physical

facilities like data centers, are categorized as large-scale systems. In these contexts, the centrality assumption becomes untenable due to the absence of a centralized information hub and limitations in centralized computing capabilities. Factors such as the system's high dimensionality, uncertainties, and communication delays in the network, along with the geographical dispersion of components, compounded by the rapid advancements in microprocessor technologies, necessitate a transition from centralized control to decentralized decision-making and distributed computations [10, 15]. Among various advanced control techniques suitable for large-scale systems, model predictive control (MPC) stands out for its capacity to manage multi-variable control problems with constraints. However, centralized MPC is ill-suited for large-scale networked systems due to scalability issues and the challenges associated with maintaining global models [10]. Consequently, the development of decentralized and distributed MPC algorithms has naturally evolved to tackle these obstacles [7, 15]. These approaches involve breaking down the original optimization problem from the centralized controller into several smaller optimization problems, each addressed by separate decentralized or distributed local controllers. As a result, decentralized and distributed control structures offer a practical means of disentangling large-scale processes and reducing the computational burden of centralized control problems. This is achieved by deploying multiple MPCs that collaborate iteratively to achieve a common control objective applicable to the entire system [14, 51].

Addressing time delays poses another challenge in establishing controlling processes. These delays stem from various sources, including the computation of control inputs, communication lags during signal transfer, material transportation dynamics within the process network, and control actuator dynamics. Advancements in networked communication have streamlined connectivity and

data transfer in cyber-physical systems and have minimized communication delays. However, delays arising from control actuator dynamics remain a challenge. These delays cannot be mitigated solely by reducing control input computation times or enhancing material transport and networked communication speed. Therefore, it is crucial to employ appropriate control strategies, such as integrating predictors within the controller design, to address these issues effectively [79]. Another common challenge involves the acquisition of sensor measurements for all states in large processes, which can be cost-prohibitive and impractical due to installation complexities. Consequently, extensive research has been devoted to state estimation techniques, enabling real-time prediction of unmeasured states through deterministic and stochastic methods. Notably, the extended Kalman Filter (EKF) and extended Luenberger observer (ELO) are widely utilized for nonlinear processes [48, 89].

Improving the economic performance, efficiency, and adaptability to fluctuating economics in real-time has long been a central research focus in dynamic process optimization. Studies in chemical process control indicate that many industrial processes can enhance profitability by adopting time-varying operations over static steady-state methods [6, 29]. This trend has given rise to economic model predictive control (EMPC), enabling the dynamic optimization of economic cost functions while ensuring stability constraints are met. The impact of fluctuating energy costs, commodity prices, currency values, interest rates, logistics expenses, and market dynamics is substantial across global industries. By integrating real-world economic variations, EMPC systems can yield superior results, highlighting the critical role of dynamic optimization techniques in maintaining competitiveness amid unpredictable conditions.

1.2 Background

Numerous real-life instances underscore the criticality of cybersecurity in networked cyber-physical systems and SCADA environments. In 2010, the Stuxnet worm targeted SCADA systems in Iranian nuclear facilities, causing centrifuges to burn out [47]. Similarly, in 2015, hackers compromised SCADA systems, leading to widespread power outages in the Ukrainian power grid [45]. More recently, in May 2021, the Colonial Pipeline suffered a ransomware attack, resulting in disruptions to fuel supply and significant economic impact [78, 82]. Ongoing research focuses on integrating encryption into linear model predictive control systems [18, 20]. Additionally, a comprehensive study explored the design of encrypted model predictive control for nonlinear processes and the impact of quantization on system performance [81]. However, the implementation of encrypted model predictive control in nonlinear processes with plant/model mismatch remains unexplored. Further investigation is needed to understand the effects of quantization and compare the time required for quantization-related operations with encryption-decryption.

To implement encrypted model predictive control systems in nonlinear processes, it is essential that nonlinear computations occur in plaintext after decryption. Current research generally assumes a cyber-secure decryption environment [39, 40, 81], which may not always be guaranteed. In a cyber-vulnerable environment, decrypted data can be manipulated during a cyberattack. Thus, there is a need for a control architecture that ensures cyber-resilient operation even in non-secure decryption settings. Additionally, exploring the integration of encrypted control with machine learning-based cyberattack detection is crucial. Moreover, developing encrypted control architectures that integrate linear and nonlinear control systems using different encryption methods could

bolster cybersecurity and merit further investigation.

Chemical process operations heavily rely on automated control systems, necessitating the development of model predictive control (MPC) to address multivariable interactions and input/state constraints. However, advancements in sensor technology and network-based communication introduce complexities by increasing the number of decision variables, state variables, and measurement data. This escalation in complexity can lead to longer computation times when using centralized MPC, especially for large-scale systems like power distribution grids, mechanical systems, chemical processes, supply chains, and urban traffic networks. Simply relying on faster computers with large memory is insufficient to address these challenges [7]. In response, distributed model predictive control systems have emerged, employing multiple controllers with inter-controller communication to cooperatively calculate control actions and achieve plant objectives [15, 74, 80]. Decentralized control architectures have also been proposed, where controllers independently compute control inputs without inter-controller communication [33, 56, 76]. However, despite their potential, the application of these approaches to nonlinear process systems with encrypted communication remains largely unexplored. Ensuring data confidentiality in large-scale systems while enhancing computation efficiency is crucial, and exploring the application of decentralized or distributed MPCs with encrypted communication in realistic scenarios involving nonlinear processes with state and input delays or limited state measurements is essential. Addressing these challenges will require additional provisions and can serve as a promising area for future exploration. Moreover, conducting a comprehensive comparison with quantitative metrics regarding the closed-loop performance and computational burden of encrypted centralized, decentralized, and distributed MPC systems can offer valuable insights for selecting the most suitable

control system.

Nonlinear model predictive control frameworks have been proposed, including two-tier control architectures where linear and nonlinear controllers are integrated into a single control system, with linear controllers computing control inputs in an encrypted space [41]. In such applications, encrypted control systems operate independently in a decentralized manner. However, these strategies can also be combined in a sequential framework, where a nonlinear controller computes operating points in plaintext, encrypts them, and passes them to another control layer, which tracks the set-points by computing control inputs in an encrypted space. Moreover, encrypted control frameworks can be designed to perform dynamic economic optimization and account for fluctuating economics in real-time to maximize the economic performance of nonlinear processes. Such multipurpose control systems that enhance both economic performance and cybersecurity are essential to maintain competitiveness and cybersecurity. Additionally, cyberattack detection and reconfiguration schemes must be integrated to ensure cyber-resilient operation in case of cyberattacks. While previous research has included detection [2, 26, 60, 64], integrating these methods with reconfiguration mechanisms into the new control systems as described above can lead to cyber-resilient operation.

1.3 Thesis objectives and structure

This thesis presents new control architectures that use encrypted communication to tackle the challenges faced by modern control systems, including operational safety, cybersecurity, and large-scale process control. The thesis provides theoretical analyses of these control architectures and

demonstrates their applications in nonlinear chemical process examples. The objectives of this thesis can be summarized as follows:

1. To present a centralized encrypted Lyapunov-based MPC for a nonlinear chemical process network, and analyze the effect of quantization on closed-loop performance and computation burden.

2. To integrate nonlinear and linear control systems in an encrypted two-tier control framework with machine-learning-based detection algorithms for enhanced cybersecurity.

3. To develop an encrypted decentralized model predictive control scheme for nonlinear time-delay systems with rigorous theoretical analysis on their closed-loop stability properties.

4. To create encrypted distributed model predictive control systems with extended Luenberger observer-based state estimations for nonlinear processes when only partial state measurements are available.

5. To formulate an encrypted two-layer control framework to maximize economic performance while addressing fluctuating real-world economic with cyberattack resilient operation.

The remainder of this thesis is organized as follows: Chapter 2 focuses on developing and applying Encrypted Lyapunov-based Model Predictive Control (LMPC) in a non-linear chemical process network for ethylbenzene production. The network, governed by a non-linear dynamic model, comprises two continuously stirred tank reactors that are connected in series and is simulated using Aspen Plus Dynamics. For enhancing system cybersecurity, the Paillier Cryptosystem is employed for encryption-decryption operations in the communication channels between the sensor-controller

and controller-actuator, establishing a secure network infrastructure. Cryptosystems generally require integer inputs, necessitating a quantization parameter d, for quantization of real-valued signals. We utilize the quantization parameter to quantize process measurements and control inputs before encryption. Through closed-loop simulations under the encrypted LMPC scheme, where the LMPC uses a first-principles non-linear dynamical model, we examine the effect of the quantization parameter on the performance of the controller and the overall encryption to control input calculation time. We illustrate that the impact of quantization can outweigh those of plant/model mismatch, showcasing this phenomenon through the implementation of a first-principles-based LMPC on an Aspen Plus Dynamics process model. Based on the findings, we propose a strategy to mitigate the quantization effect on controller performance while maintaining a manageable computational burden on the control input calculation time.

Chapter 3 presents an encrypted two-tier control architecture integrated with a machine learning (ML) based cyberattack detector to enhance the operational safety, cyber-security, and performance of nonlinear processes. The upper tier of this architecture employs an encrypted nonlinear Lyapunov-based model predictive controller (LMPC) to enhance closed-loop performance, while the lower tier utilizes an encrypted set of linear controllers to stabilize the process. Encrypted signals from the sensors are decrypted at the upper tier for plain text control input computation, while the lower tier computes control inputs in an encrypted space, due to its exclusive use of linear operations. While this design enhances closed-loop performance, it exposes the upper tier to potential cyberattacks. To mitigate this risk, an ML-based detector is developed in the form of a feed-forward neural network, utilizing sensor-derived data for attack detection. Upon attack detection, the control system logic deactivates the performance-enhancing upper tier and relies solely

on the cybersecure lower tier for system stabilization. The chapter also includes a comprehensive stability analysis of the two-tier control structure, establishing error bounds related to quantization and sample-and-hold controller implementations. The proposed control framework can be extended to any nonlinear process that is controlled by a combination of linear and nonlinear controllers to enhance the system cybersecurity. Guidelines such as quantization parameter selection, cyberattack detector development, and sampling time criteria are included to facilitate practical implementation. Simulation results of a nonlinear chemical process network demonstrated the robustness of the encrypted control architecture and cyberattack detector, as well as its ability to detect previously unseen attack patterns.

Chapter 4 focuses on enhancing the operational safety, cybersecurity, computational efficiency, and closed-loop performance of large-scale nonlinear time-delay systems. This is achieved by employing a decentralized model predictive controller (MPC) with encrypted networked communication. Within this decentralized setup, the nonlinear process is partitioned into multiple subsystems, each controlled by a distinct Lyapunov-based MPC. These controllers take into account the interactions between subsystems by utilizing full state feedback, while computing the control inputs only corresponding to their respective subsystem. To address the performance degradation associated with input delays, we integrate a predictor with each LMPC to compute the states after the input delay period. The LMPC model is initialized with these predicted states. To cope with state delays, the LMPC model is formulated using differential difference equations (DDEs) that describe the state-delays in the system. Further, to enhance cybersecurity, all signals transmitted to and received from each subsystem are encrypted. A stability analysis is carried out for the encrypted decentralized MPC when it is utilized in a time-delay system. Bounds are set up for

the errors arising from encryption, state-delays, and sample-and-hold implementation of the controller. Guidelines are established to implement this proposed control structure in any nonlinear time-delay system. The simulation results, conducted on a nonlinear chemical process network, illustrate the effective closed-loop performance of the decentralized MPCs alongside the predictor with encrypted communication when dealing with input and state delays in a large-scale process.

Chapter 5.1 and Chapter 5.2 explore the design of encrypted distributed MPC systems for nonlinear processes. Firstly, a distributed model predictive controller (DMPC) is utilized to partition the process into multiple subsystems, each controlled by a distinct Lyapunov-based MPC (LMPC). To consider the interactions among different subsystems, each controller receives and shares control inputs computed for its subsystem. As full state feedback is unavailable, we integrate an extended Luenberger observer with each LMPC, initializing the LMPC model with complete state estimate information provided by the observer. Furthermore, to enhance cybersecurity, wireless signals received and transmitted by the controllers are encrypted. Guidelines are established to implement this proposed control structure in any large-scale nonlinear chemical process network. Simulation results, conducted on a specific nonlinear chemical process network, demonstrate the effective closed-loop performance of the encrypted DMPC with state estimation, utilizing partial state feedback with sensor noise. This is followed by a comprehensive comparison of the closed-loop performance, control input computational time, and suitability of encrypted centralized, decentralized, and distributed MPC frameworks. Secondly, an encrypted iterative DMPC with encrypted communication links between sensors, actuators, and control input computing units is presented. Through a comprehensive stability analysis of the encrypted iterative DMPC, bounds are established on errors arising from encrypted communication links,

disturbances, and the sample-and-hold implementation of controllers. Practical aspects such as reducing data encryption time by appropriate key length choices, sampling interval criterion, and quantization parameter selection are discussed. Simulation results of the proposed control scheme applied to a nonlinear chemical process, showcase its effective closed-loop performance in the presence of sensor noise and process disturbances. Specifically, a non-Gaussian noise distribution is obtained from an industrial data set and added to the state measurements to justify the practical effectiveness of the proposed approach.

Chapter 6 proposes a two-layer framework to maximize economic performance through dynamic process economics optimization while addressing fluctuating real-world economics and enhancing cyberattack resilience via encryption in the feedback control layer for nonlinear processes. The upper layer employs a Lyapunov-based economic model predictive control scheme, receiving updated economic information for each operating period, while the lower layer utilizes an encrypted linear feedback control system. Encrypted state information is decrypted in the upper layer to determine the economically optimal dynamic operating trajectory through nonlinear optimization. Conversely, the lower layer securely tracks this trajectory in an encrypted space without decryption. To mitigate the cyber vulnerability of the upper layer, we integrate a cyberattack detector that utilizes sensor-derived data for attack detection. We quantify the errors emanating from quantization, disturbances, and sample-and-hold controller implementation. Simulation results of a nonlinear chemical process highlight the robustness and economic benefits of the new control architecture.

Chapter 7 summarizes the main results of the thesis.

Chapter 2

Encrypted Model Predictive Control of a Nonlinear Chemical Process Network

2.1 Introduction

With the rapid advancement of technology and the increasing integration of devices, networked cyber-physical systems, particularly those utilizing SCADA (Supervisory Control and Data Acquisition) technology, have become integral components of critical infrastructure across industries such as energy, water, transportation, and manufacturing. These systems enable efficient monitoring, control, and automation of complex processes, enhancing productivity and operational efficiency. However, the increased connectivity and integration of SCADA systems with corporate networks and the Internet have exposed them to potential cyber threats. A breach or compromise in these systems can have severe consequences, including disruption of essential services, physical damage, financial losses, and even threats to public safety. Recent advances in cyberattack techniques and the growing sophistication of threat actors have further highlighted the criticality

of implementing robust cybersecurity measures.

Various real-world examples underscore the importance of cybersecurity in networked cyber-physical systems and SCADA environments. For instance, the Stuxnet worm, discovered in 2010, specifically targeted SCADA systems in Iranian nuclear facilities. Stuxnet infiltrated Iranian PLCs (Programmable Logic Controllers), gathering data about industrial systems and causing the fast-spinning centrifuges to burnout [47]. Another notable incident is the Ukrainian power grid cyber-attack in 2015, where hackers successfully compromised SCADA systems, leading to widespread power outages affecting thousands of people. In a recent incident in May 2021, the Colonial Pipeline, a major fuel pipeline operator in the United States, fell victim to a ransomware attack. The attackers infiltrated Colonial Pipeline's network through the DarkSide ransomware. They encrypted the company's systems and demanded a ransom payment in exchange for the decryption keys. As a result, Colonial Pipeline shut down its operations, leading to disruptions in fuel supply and causing a significant economic impact.

Despite significant advancements in addressing cybersecurity challenges within the information technology (IT) domain, the operational technology (OT) domain is still catching up in terms of progress. IT primarily focuses on the software component of systems, encompassing network infrastructure and data management. In contrast, OT ensures the smooth operation of critical infrastructure, including power grids, smart meters, and distribution systems. Notably, cyberattacks targeting OT systems tend to have more severe and far-reaching consequences compared to those in IT. These attacks can lead to outcomes such as shutdowns, outages, leakages, and even explosions. Consequently, standards development organizations like the National Institute of Standards and Technology (NIST) [9] have devised essential cybersecurity research roadmaps. These roadmaps

serve as frameworks designed to identify and mitigate the impact of cyber-attacks, thereby exerting a notable influence on the security protocols adopted across various industries.

While significant research efforts continue to focus on diverse domains, such as the creation of machine learning-based cyber-attack detectors [1, 36, 66, 84], the design of backup controllers in a two-tier safety-performance control architecture [13], the recovery of process states following a cyber-attack [87], and the development of cyberattack-resilient controllers [24, 25], one critical and fundamental research issue remains unresolved: the establishment of universally implementable secure data transmission lines in any cyber-physical networked system, without requiring controller modifications, installation of backup control systems, development of system-specific detection mechanisms, or tailor-made solutions for individual platforms. A promising solution to address this issue is utilizing an encrypted control system. This approach offers a versatile and effective solution for enhancing data security and confidentiality. It can be easily implemented across various systems without necessitating system-specific modifications, thereby addressing the fundamental challenge of secure data transmission in networked systems.

Regarding encrypted control, extensive research has been conducted in the field of linear control systems, with control computations performed in a fully encrypted space. The fundamental concept behind such systems is multiplicative homomorphism, which enables multiplication operations to be executed in an encrypted medium using complex cryptosystems like the ElGamal [27]. However, such operations in an encrypted space can be computationally demanding and not applicable to systems governed by complicated non-linear dynamics where non-linear controllers may be needed, limiting their widespread adoption. Alternatively, a more viable approach could involve using encryption to secure data transmission lines. The data collected by the sensors

can be encrypted, subsequently transferred, and decrypted at the controller, which can be isolated and fortified against potential security breaches. Therefore, within the context of this research, we consider that the edge computer, responsible for executing controller computations within a SCADA architecture, operates within a completely secure cyber-physical setting due to encryption of the sensor-to-controller and controller-to-actuator signals. Specifically, in our formulation, the controller can compute the control action in plaintext, eliminating the need for convoluted calculations in an encrypted space. Subsequently, the control action can undergo encryption before transmission to the actuator, where the encrypted control action is decrypted and executed. This method avoids computationally demanding operations in an encrypted space and is effectively implementable in systems employing advanced process control schemes for non-linear systems, such as model predictive control (MPC).

Since its inception, the chemical industry has extensively adopted model predictive control due to its effectiveness in achieving closed-loop stability, optimizing key performance metrics, its capability to handle multiple inputs and outputs, and accommodate constraints on system states and inputs. These benefits arise from employing a mathematical model of the system to predict future behavior and optimize control inputs accordingly. However, implementing MPC necessitates decryption at the controller to obtain the essential information required for prediction and optimization. In an industrial setting, an edge computer, accessible remotely by the sensors and actuators through the network, can perform non-linear MPC computations. The objective is to utilize encryption techniques to establish secure connections between the sensors-edge computer and edge computer-actuators. The referenced work [81] provides a comprehensive exploration of the design of an encrypted model predictive control framework, as well as the influence of quantization

on system performance. Building upon that foundation, in this work, we go a step further by implementing the encrypted Lyapunov-based model predictive control (LMPC) scheme in a large-scale chemical process network used for ethylbenzene production, using an Aspen Plus Dynamics based process model in conjunction with a first-principles based LMPC to showcase that the influence of quantization can surpass the impact of plant/model mismatch. Moreover, the study conducts a comprehensive and innovative investigation to assess how encryption-decryption affects the computation time required for computing the control action. By thoroughly examining the impact of the quantization parameter selected for encryption on the computation time, this research aims to provide new perspectives and deeper insights into the practical implications of data encryption. To our knowledge, prior investigations have not explored the implications of an encrypted MPC scheme in the aforementioned domains.

To apply the encrypted LMPC, we develop two distinct non-linear dynamical models: one utilizing Aspen Plus Dynamics V12 and the other based on first-principles modeling fundamentals. In Section 2.4, we conduct closed-loop simulations for the Aspen Plus Dynamics model, employing the first-principles model-based encrypted LMPC for various quantization parameters. Further, we investigate the impact of these parameters on controller performance and put forth a proposal to mitigate quantization errors and their effects on controller performance. Additionally, in Section 2.5, we explore the influence of encryption-decryption on the total control input calculation time. Expanding on the previous recommendation, we provide clear guidance on implementing the encrypted LMPC approach. This implementation ensures a feasible computation time for control action computation (with encryption) while establishing secure communication pathways between the sensor-controller and controller-actuator components, without compromising the performance

of the controller.

2.2 Preliminaries

2.2.1 Notation

The Euclidean norm of a vector is denoted by the symbol $\|\cdot\|$. The notation x^T represents the transpose of the vector x. The standard Lie derivative $L_f V(x)$ is defined as the partial derivative of the function $V(x)$ with respect to x multiplied by the vector field $f(x)$, $L_f V(x) := \frac{\partial V(x)}{\partial x} f(x)$. The sets $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ refer to the sets of real numbers, integers, and natural numbers, respectively. Additionally, $\mathbb{Z}_M$ and $\mathbb{Z}_M^*$ represent the additive and multiplicative groups of integers modulo M, respectively.

The set subtraction operation is denoted by "\", meaning that $A \backslash B$ represents the set of elements in A that are not in B. A function $f(\cdot)$ is said to be of class $\mathcal{C}^1$ if it is continuously differentiable in its domain. A continuous function $\alpha : [0, a) \to [0, \infty)$ is considered to be in the class $\mathcal{K}$ if it is strictly increasing and only evaluates to zero at zero. The function $\gcd(i, j)$ denotes the greatest common divisor, which returns the largest positive integer that divides both i and j without leaving a remainder. On the other hand, $\mathrm{lcm}(i, j)$ represents the least common multiple of the integers i and j.

2.2.2 Class of Systems

In this work, we primarily focus on a specific category of systems known as nonlinear continuous-time systems with multiple input and multiple outputs (MIMO). These systems represent a set of first-order ordinary differential equations (ODEs) that exhibit nonlinear behavior. The general representation of these systems is:

$$\dot{x} = F(x, u) = f(x) + g(x)u \tag{2.1}$$

The system is described by a state vector $x = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$ and a control input vector $u \in \mathbb{R}^m$. The inputs applied to the system are subject to certain bounds, defined by the set $U \subset \mathbb{R}^m$, where $U := \{u \in U | u_{\min,i} \leq u_i \leq u_{\max,i}, \forall; i = 1, 2, \cdots, m\}$. The values $u_{\min,i}$ and $u_{\max,i}$ represent the minimum and maximum limits for each manipulated input, respectively. The functions $f(\cdot)$ and $g(\cdot)$ are assumed to be sufficiently smooth vector and matrix functions, respectively. For the sake of simplicity and without sacrificing the general applicability, we make the assumption that $f(0) = 0$, thereby considering the origin as a steady state of the nonlinear system described by the Eq. (2.1). For convenience, we set the initial time to zero throughout the paper ($t_0 = 0$). In addition, we introduce some notation: the space of continuous functions that map the interval $[a, b]$ to $\mathbb{R}^n$ is denoted by $C([a, b], \mathbb{R}^n)$. We also define the set $S(\Delta)$ as the collection of piece-wise constant functions with a period of Δ.

2.2.3 Achieving Stability through Lyapunov-based Feedback Control

We assume the existence of a feedback controller denoted as $u = \Phi(x) \in U$ to achieve exponential stability at the origin within the system described by the Eq. (2.1). This exponential stability is characterized by the presence of a continuously differentiable control Lyapunov function denoted as $V(x)$, satisfying the following inequalities for all x within an open neighborhood D around the origin [85, 86]:

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

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

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

where c_1, c_2, c_3 and c_4 are positive constants. The method presented in the referenced work, [49] offers an approach to construct a stabilizing controller that satisfies the desired criteria. For the nonlinear system of Eq. (2.1), the closed loop stability region is characterized as a level set of the Lyapunov function V. This stability region Ω_ρ is defined as $\Omega_\rho := \{x \in D | V(x) \leq \rho\}$, where $\rho > 0$.

2.2.4 Paillier cryptosystem

In this research article, we utilize the Paillier cryptosystem [67] to apply encryption and decryption to process measurements (represented as x) and control inputs (represented as u). The Paillier cryptosystem is a partially homomorphic encryption scheme that enables performing addition operations within the encrypted message space. However, the primary rationale for utilizing the Paillier

cryptosystem in this paper is its computational efficiency compared to other cryptosystems, such as ElGamal or AES, rather than its partial homomorphic property. Like most cryptosystems, the Paillier cryptosystem operates by encrypting plaintext data presented in the form of non-negative integers. The encryption process commences with the generation of public and private keys. The public key is used to encrypt integer messages and produce ciphertexts. Conversely, the private key decrypts the ciphertexts and recovers the original integer messages. The generation of the public and private keys in the Paillier cryptosystem follows a specific set of steps:

1. Select two large random prime integers (p and q) satisfying the condition $\gcd(pq, (p-1)(q-1)) = 1$.

2. Calculate the product of these integers, denoted by $M = pq$.

3. Select a random integer g such that, $g \in \mathbb{Z}^*_{M^2}$ where $\mathbb{Z}^*_{M^2}$ is the multiplicative group of integers modulo M^2.

4. Calculate $\lambda = \mathrm{lcm}(q-1, p-1)$.

5. Define $L(x) = (x-1)/M$.

6. Check the existence of the following modular multiplicative inverse:
   $u = (L(g^\lambda \bmod M^2))^{-1} \bmod M$.

7. If the inverse does not exist, return to step 3 and select an alternative value for g. In the event that the inverse does exist, we obtain the public key (M, g) and the private key (λ, u).

After obtaining the keys, we distribute the public key to the intended recipients that perform the encryption process. Similarly, we share the private key exclusively with the authorized recipients

responsible for decrypting the data. The process of encryption-decryption consists of the following steps:

$$E_M(m, r) = c = g^m r^M \bmod M^2 \tag{2.3}$$

where $r \in \mathbb{Z}_M$ is a random integer and c is the ciphertext obtained after encryption of m. The decryption process of the ciphertext $c \in \mathbb{Z}_{M^2}$, is performed as follows:

$$D_M(c) = m = L(c^\lambda \bmod M^2)u \bmod M \tag{2.4}$$

2.2.5 Quantization

In order to utilize the Paillier cryptosystem, it is necessary to represent the input data to be encrypted as natural numbers. However, it's important to note that the signal measurements provided before encryption are typically in the form of floating-point numbers. Consequently, a mapping procedure becomes essential to convert these floating-point numbers into elements within the set $\mathbb{Z}_M$. This procedure involves quantization, where a quantization parameter denoted by d is chosen to perform the quantization operations [19].

To achieve this objective, we adopt signed fixed-point numbers in binary representation. The quantization parameters l_1 and d refer to the total number of bits and the number of fractional bits, respectively. Using these quantization parameters, we construct a set denoted as $\mathbb{Q}_{l_1,d}$. This set encompasses rational numbers ranging from -2^{l_1-d-1} to $2^{l_1-d-1} - 2^{-d}$, with each rational number separated by a resolution of 2^{-d}. A rational number q belonging to the set $\mathbb{Q}_{l_1,d}$ can be expressed as follows: $q \in \mathbb{Q}_{l_1,d}$, where $\exists \beta \in \{0,1\}^{l_1}$ and $q = -2^{l_1-d-1}\beta_{l_1} + \sum_{i=1}^{l_1-1} 2^{i-d-1}\beta_i$. To map a real

number data point a to the set, $\mathbb{Q}_{l_1,d}$ we employ the function $g_{l_1,d}$ given by the equation:

$$\begin{aligned} g_{l_1,d} &: \mathbb{R} \to \mathbb{Q}_{l_1,d} \\ g_{l_1,d}(a) &:= \arg\min_{q \in \mathbb{Q}_{l_1,d}} |a - q| \end{aligned} \tag{2.5}$$

This function allows us to determine the closest quantized rational number to a given real number data point. Following this quantization step, the quantized data is mapped to a set of integers using bijective mapping denoted as $f_{l_2,d}$ [19]. This mapping ensures that the quantized data is transformed into a subset of the message space $\mathbb{Z}_M$. The bijective mapping can be defined as:

$$\begin{aligned} f_{l_2,d} &: \mathbb{Q}_{l_1,d} \to \mathbb{Z}_{2^{l_2}} \\ f_{l_2,d}(q) &:= 2^d q \bmod 2^{l_2} \end{aligned} \tag{2.6}$$

The encryption process involves encrypting integer plaintext messages using the set, $Z_{2^{l_2}}$ and the resulting ciphertexts can be decrypted back into the same set $Z_{2^{l_2}}$. Once the controller and actuator receive the encrypted signals, the ciphertexts undergo decryption to extract integer plaintext messages that represent quantized states and inputs, respectively. Consequently, it becomes essential to remap these decrypted plaintext messages back to the set $\mathbb{Q}_{l_1,d}$. The inverse mapping, denoted as $f^{-1}_{l_2,d}$, is defined as follows:

$$f^{-1}_{l_2,d} : \mathbb{Z}_{2^{l_2}} \to \mathbb{Q}_{l_1,d} \tag{2.7}$$

$$f^{-1}_{l_2,d}(m) := \frac{1}{2^d} \begin{cases} m - 2^{l_2} & \text{if } m \geq 2^{l_2 - 1} \\ m & \text{otherwise} \end{cases} \tag{2.8}$$

To demonstrate encryption and decryption, we can refer to Figure 2.1. For this example, the chosen

quantization parameter, total number of bits, and bijective mapping parameter are: $d = 3$, $l_1 = 18$, and $l_2 = 30$. Let's consider the rational number $a = -1.31752$ which is the input data to be encrypted to illustrate the encryption-decryption process and the effect of quantization.

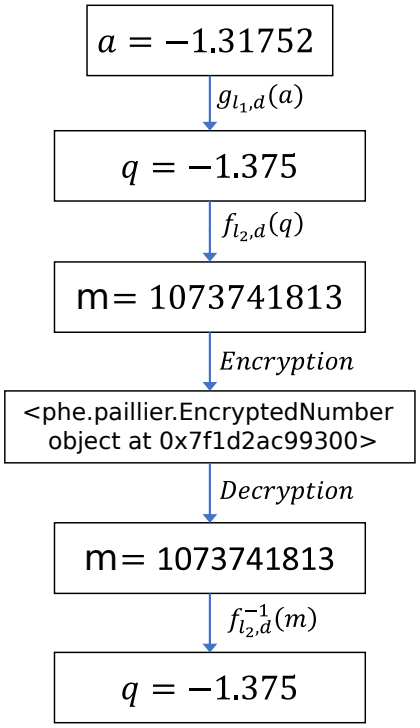

Figure 2.1: Illustration of encryption-decryption applied to a floating-point real number.

2.3 Design of the Encrypted MPC

In the envisioned closed-loop architecture of the encrypted MPC, as depicted in Figure 2.2, the sensor signals $x(t)$ are subjected to encryption before being sent to the model predictive controller (MPC). After obtaining the encrypted data, it undergoes decryption, resulting in quantized states $\hat{x}(t)$. These quantized states serve as the initial values for the plant model within the MPC at time t. The MPC subsequently computes optimized inputs $u(t)$, which are encrypted prior to transmission

to the actuator. After the actuator receives the encrypted signals as input, the encrypted input is decrypted, leading to a quantized input, $\hat{u}(t)$ that is applied to the process.

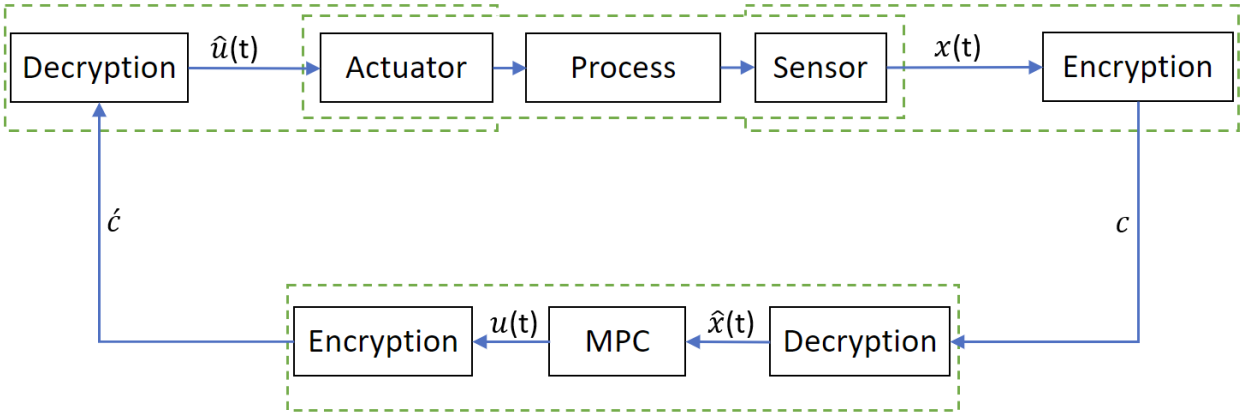

Figure 2.2: Illustration of the data transfer process in an encrypted MPC system.

The above closed-loop design introduces two sources of errors. Firstly, a quantization error in the sensor-MPC communication link, resulting from the mapping of the state data from $\mathbb{R}$ to $\mathbb{Q}_{l_1,d}$. Additionally, the MPC-actuator communication link introduces an input quantization error caused by the conversion of input data from the set of real numbers $\mathbb{R}$ to $\mathbb{Q}_{l_1,d}$. These quantization errors are bounded and can be characterized by the mapping equation of Eq. (2.5), ensuring that:

$$|x(t) - \hat{x}(t)| \leq 2^{-d-1} \tag{2.9a}$$

$$|u(t) - \hat{u}(t)| \leq 2^{-d-1} \tag{2.9b}$$

where d is the quantization parameter used for mapping in Eq. (2.5). Firstly, taking into account the impact of quantization-induced input errors, the dynamical model of the MPC employs a nonlinear

system, represented by Eq. (2.1), which can be expressed as follows:

$$\begin{aligned} \dot{x} = F(x, \hat{u}) &= f(x) + g(x)\hat{u} \\ &= f(x) + g(x)(u + e) \end{aligned} \tag{2.10}$$

where $e = \hat{u}(t) - u(t)$ and

$$|e| \leq 2^{-d-1} \tag{2.11}$$

Secondly, an error in the control input, $u = \Phi(x) \in U$, will emanate as the MPC receives $\hat{x}$ instead of the actual state x. This error will be bounded by the underlying equation, where $L_1 > 0$:

$$|\Phi(\hat{x}) - \Phi(x)| \leq L_1|\hat{x} - x| \leq L_1 2^{-d-1} \tag{2.12}$$

Reference [81] discusses and establishes the stability of the proposed control loop with encrypted data transfer, providing assurance for the closed-loop system stability even in the presence of encryption, under certain conditions.

Remark 2.1. *The error in the quantization operation occurs when the target value to be quantized is not found exactly in the set $\mathbb{Q}_{l_1,d}$, which consists of quantized values with a certain resolution determined by the quantization parameter, denoted as d. The resolution between elements in this set is given by 2^{-d}. To determine the upper bound of the error, let's focus on a specific value, denoted as x_1, that needs to be quantized. We assume that x_1 falls within the range of y_1 and $y_1 + 2^{-d}$, where y_1 and $y_1 + 2^{-d}$ represent quantized values in the set $\mathbb{Q}_{l_1,d}$. The quantization process involves comparing the distance between x_1 and y_1 with the distance between x_1 and $y_1 + 2^{-d}$. If the distance between x_1 and y_1 is smaller than the distance between x_1 and $y_1 + 2^{-d}$,*

then x_1 is mapped to y_1. Otherwise, it is mapped to $y_1 + 2^{-d}$. The error in quantization is then bounded by half the resolution, which is equal to $|y_1 + 2^{-d} - y_1|/2 = 2^{-d-1}$. This implies that the maximum difference between the quantized value $\hat{x}_1$, and the actual value x_1, is 2^{-d-1}.

2.3.1 Encrypted Lyapunov-based MPC

This section presents a formulation of feedback MPC for the closed-loop design of the nonlinear system described by Eq. (2.1), considering secure sensor-controller and controller-actuator communication links. Control actions will be applied to the nonlinear system using a sample-and-hold approach with a sampling period of Δ [35, 59]. The proposed MPC formulation is outlined as follows:

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L_2(\tilde{x}(t), u(t)) dt \tag{2.13a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F(\tilde{x}(t), u(t)) = f(\tilde{x}) + g(\tilde{x})u \tag{2.13b}$$

$$u(t) \in U, \ \forall \ t \in [t_k, t_{k+N}) \tag{2.13c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{2.13d}$$

$$\dot{V}(\hat{x}(t_k), u) \leq \dot{V}(\hat{x}(t_k), \Phi(\hat{x}(t_k))), \ \text{if} \ \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{\min}} \tag{2.13e}$$

$$V(\tilde{x}(t)) \leq \rho_{\min}, \ \forall \ t \in [t_k, t_{k+N}), \ \text{if} \ \hat{x}(t_k) \in \Omega_{\rho_{\min}} \tag{2.13f}$$

Within the framework of the Lyapunov-based MPC, referred to as LMPC, the predicted state trajectory is represented as $\tilde{x}$, the sampling time is denoted by Δ, and the prediction horizon encompasses a number of sampling periods indicated by N. The LMPC algorithm computes the optimal input sequence $u^*(t|t_k)$ for the entire prediction horizon $t \in [t_k, t_{k+N})$. The first input of this se-

quence is subsequently transmitted to the actuator for application to the system within the interval $t \in [t_k, t_{k+1})$.

In the encrypted LMPC design, the MPC uses quantized states $\hat{x}$ for predicting the state trajectory, Eq. (2.13a) integrates the cost function over the entire prediction horizon, and computes the optimized control inputs for the entire prediction horizon. However, the actuator only applies the control inputs corresponding to the first prediction horizon and repeats this process at each sampling instance. Eq. (2.13b) represents the dynamic system model used by the LMPC. Eq. (2.13c) represents the constraints imposed on the control inputs. The constraint in Eq. (2.13d) initializes the plant model described in Eq. (2.13b) with quantized states. If the state $x(t_k)$ at time t_k lies within the set $\Omega_{\hat{\rho}} \setminus \Omega_{\rho_{\min}}$, where $\rho_{\min}$ represents a level set of V in proximity to the origin, the Lyapunov constraint outlined in Eq. (2.13e) steers the closed-loop state $x(t_k)$ of the nonlinear system presented in Eq. (2.10) towards the origin. Once the closed-loop state $x(t_k)$ enters the region $\Omega_{\rho_{\min}}$, the constraint specified in Eq. (2.13f) ensures that this state remains within $\Omega_{\rho_{\min}}$ throughout the entire prediction horizon.

2.4 Application to a chemical process operating at an unstable steady-state using Aspen Plus simulator

In this section, we demonstrate the application of the proposed encrypted LMPC to a large-scale chemical process. To begin, we construct two dynamic models for a chemical process. We develop the first dynamic model using Aspen Plus Dynamics V12, while the second model is based on first-principles modeling fundamentals. Aspen Plus Dynamics is a high-fidelity software that can be

used for detailed dynamic simulation of chemical processes in an operating region around a stable or unstable steady-state, which is not possible in steady-state simulation software for chemical processes, and hence, can be considered as the closest representation of the actual process dynamic behavior. Furthermore, first-principles based MPC computations can be done on a computer in SCADA systems using Python. As a first-principles model can be derived for most processes even in the absence of data and be simulated readily with available solvers, the Aspen Plus Dynamics model and first-principles based Python code can be considered as a "standard metric" to quantify and analyze specific aspects of MPC. In this work, we use a distinct model to simulate the chemical process from the model incorporated into the LMPC to demonstrate the impact of quantization and compare it with plant/model mismatch. We design both models without any input or state delays. Subsequently, closed-loop simulations are performed in the Aspen Plus Dynamics model using the first-principles model-based LMPC. Finally, we replace the LMPC with an encrypted LMPC, and closed-loop simulations are conducted and discussed.

2.4.1 Process Description

The process considered is the production of Ethylbenzene (EB) from Ethylene (E) and Benzene (B) as reactive raw materials. The main reaction, labeled as "primary," is a second-order, exothermic, and irreversible reaction that occurs alongside two additional side reactions. This reaction scheme is illustrated in Eq. (2.14) and takes place in two non-isothermal, well-mixed continuous stirred

tank reactors (CSTR). The chemical reactions involved are as follows:

$$C_2H_4 + C_6H_6 \longrightarrow C_8H_{10} \quad (\text{primary}) \tag{2.14a}$$

$$C_2H_4 + C_8H_{10} \longrightarrow C_{10}H_{14} \tag{2.14b}$$

$$C_6H_6 + C_{10}H_{14} \longrightarrow 2\,C_8H_{10} \tag{2.14c}$$

The state variables are the concentration of Ethylene, Benzene, Ethylbenzene, Di-Ethylbenzene and the reactor temperature, for each $\mathrm{CSTR}_i,\ i = (1, 2)$, respectively in deviation terms that is:

$x^T = [C_{E_1} - C_{E_{1s}},\ C_{B_1} - C_{B_{1s}},\ C_{EB_1} - C_{EB_{1s}},\ C_{DEB_1} - C_{DEB_{1s}},\ T_1 - T_{1s},\ C_{E_2} - C_{E_{2s}},\ C_{B_2} - C_{B_{2s}},\ C_{EB_2} - C_{EB_{2s}},\ C_{DEB_2} - C_{DEB_{2s}},\ T_2 - T_{2s}]$

The subscript "s" denotes the steady-state value. The rate of heat removal for each reactor $[Q_1 - Q_{1s},\ Q_2 - Q_{2s}]$ are the manipulated inputs to our process, which are bounded by the closed sets $[-10^4\ \mathrm{kW},\ 2\ \mathrm{x}\ 10^3\ \mathrm{kW}]$ and, $[-1.5\ \mathrm{x}\ 10^4\ \mathrm{kW},\ 5\ \mathrm{x}\ 10^3\ \mathrm{kW}]$ respectively.

The control objective is to maintain the operation of both the CSTRs at their unstable steady-state under the encrypted LMPC using the quantized states and inputs in computation and actuation. Since the rate of heat removal for each CSTR is the manipulated input, the reactor temperature state variables are directly affected by it. However, the manipulated inputs do not directly influence the concentration states. Instead, they follow open-loop trajectories, gradually converging to their respective steady-state values as the reactor temperatures approach their steady-state values.

To identify the stability condition of the operating steady-state, we conducted an open-loop simulation in Aspen Plus Dynamics. We initiated the simulation using the steady-state values as initial conditions, and the control inputs were held constant at their respective steady-state values

(0 in deviation form) throughout the simulation. After running the simulation for 10 hours of process time, the system states converged to a distinct stable steady-state, providing clear evidence that the selected operating condition is an unstable steady-state. The main reason behind choosing this unstable steady-state as the operating condition was its ability to yield the highest amount of ethylbenzene, our desired product, at steady-state, at the outlet of the second CSTR.

2.4.2 Dynamic Model in Aspen Plus Dynamics

We develop the process model for this system using Aspen Plus and Aspen Plus Dynamics V12. These are high-fidelity simulators used for complex chemical process modeling. The two CSTRs are connected in series, such that the output of the first reactor affects the second reactor but not vice versa. Initially, the process model is created in Aspen Plus, a steady-state simulation is performed and validated by examining material and energy balances. Subsequently, dynamic simulations of the process are conducted in Aspen Plus Dynamics, enabling a thorough analysis and control of its dynamic behavior. The construction of both the steady-state and dynamic models follows the following procedure in detail:

1. Inlet stream configuration: We enter the inlet stream components, concentrations, and temperatures into Aspen Plus and supply it to each reactor through Hexane solutions with flow rates F_1 and F_2. Using Hexane ensures the inlet flows remain in the liquid phase at the feeding temperature. C_E, C_B, C_{EB}, and C_{DEB} represent the concentrations of Ethylene, Benzene, Ethylbenzene, and Di-Ethylbenzene in the inlet stream, respectively. T_i, ρ_i, V_i, are the temperature, liquid density and volume of CSTR_i, $i = 1, 2$. C_P represents the mass-

specific heat capacity of the liquid mixture, and is assumed to remain constant throughout the process in both reactors. Table 2.1 specifies the process parameters used. The subscript "o" denotes the state in the inlet stream, and "s" indicates the steady-state conditions.

2. Pressure drop selection: Valves play a crucial role in establishing a dynamic model for Aspen Plus Dynamics, as they serve as connectors between components and regulate fluid flow by controlling pressure drop across the system. A suitable pressure drop specifies the flow direction, ensuring a smooth simulation run. In our model, valves $v_1,\ v_2,\ v_3$, and v_4 are assigned pressure drops of 5, 5, 2, and 14 bars, respectively.

3. Reaction and Reactor specification: We define the reaction parameters and stoichiometry in Aspen Plus. All reactions mentioned in Eq. (2.14) are selected in the kinetic specifications of both the CSTRs. We set the initial pressure of each CSTR to 15 bar and equip them with a heating/cooling jacket to provide or remove heat at a rate denoted by Q_i, where, i represents the reactor number. The initial temperatures of the first and second CSTR are 350K and 400K, respectively. These settings ensure that the reactants and products remain in the liquid phase throughout the process. After completing the reaction specification for both reactors, we carry out a steady-state simulation.

4. Reactor Geometry: Before exporting the steady-state model from Aspen Plus to Aspen Plus Dynamics, it is necessary to define the reactor geometry. In our model, the vessels are of the vertical type with flat heads, and each CSTR has a length of ten meters.

5. Pressure Verification: To ensure the accuracy of the dynamic model, perform a pressure

check using the integrated Aspen Plus pressure checker. This step verifies that no errors arise during the dynamic process. Once the steady-state model successfully passes the pressure check, we export it to Aspen Plus Dynamics for further analysis and simulation.

6. Dynamic model initialization: Level controllers are added to each reactor to maintain them at the desired capacity. We perform a steady-state simulation to determine the steady-state values of the dynamic model. The values obtained are listed in Table 2.1. Further, we specify the initial values of the states in both reactors for the dynamic simulation. Through an initialization run, we ensure the values entered are thermo-kinetically consistent with the model specifications.

7. Manipulated Input Configuration: For external control of the manipulated variables Q_1 and Q_2 (heat duty of reactor 1 and 2, respectively) during the dynamic simulation, the heating type of the reactors is switched to constant heat duty. With these adjustments, the dynamical process model is now fully established. Figure 2.3 depicts the corresponding model flow sheet.

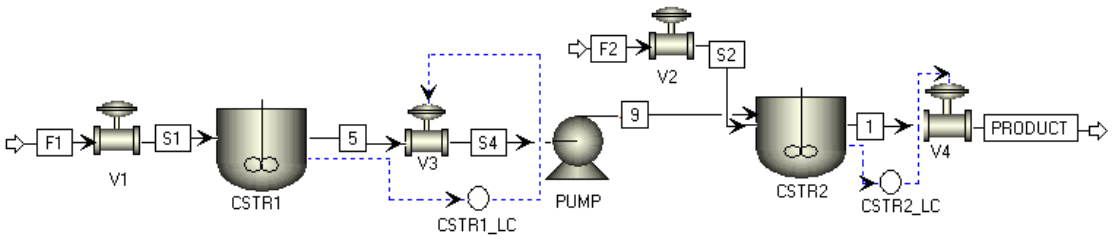

Figure 2.3: Aspen Plus Dynamics model flow sheet

Table 2.1: Parameter values, steady-state values, and model configuration of the Aspen Model

$T_{1o} = T_{2o} = 350\,\mathrm{K}$	$T_{1s} = 321.15\,\mathrm{K}$
$V_1 = V_2 = 60\,\mathrm{m}^3$	$T_{2s} = 442.99\,\mathrm{K}$
$F_1 = 43.2\,\mathrm{m}^3/\mathrm{h}$	$F_2 = 47.87\,\mathrm{m}^3/\mathrm{h}$
$C_{E_{o1}} = 4.43\,\mathrm{kmol}/\mathrm{m}^3$	$C_{E_{1s}} = 4.33\,\mathrm{kmol}/\mathrm{m}^3$
$C_{B_{o1}} = 5.54\,\mathrm{kmol}/\mathrm{m}^3$	$C_{B_{1s}} = 5.55\,\mathrm{kmol}/\mathrm{m}^3$
$C_{E_{o2}} = 4.02\,\mathrm{kmol}/\mathrm{m}^3$	$C_{E_{2s}} = 0.196\,\mathrm{kmol}/\mathrm{m}^3$
$C_{B_{o2}} = 5.02\,\mathrm{kmol}/\mathrm{m}^3$	$C_{B_{2s}} = 1.31\,\mathrm{kmol}/\mathrm{m}^3$
$C_{EB_{1s}} = 0.53\,\mathrm{kmol}/\mathrm{m}^3$	$C_{EB_{2s}} = 4.22\,\mathrm{kmol}/\mathrm{m}^3$
$C_{DEB_{1s}} = 8.76 \times 10^{-4}\,\mathrm{kmol}/\mathrm{m}^3$	$C_{DEB_{2s}} = 0.0078\,\mathrm{kmol}/\mathrm{m}^3$
$k_1 = 1.528 \times 10^6\,\mathrm{m}^3\,\mathrm{kmol}^{-1}\,\mathrm{s}^{-1}$	$E_1 = 71\,160\,\mathrm{kJ}/\mathrm{kmol}$
$k_2 = 2.778 \times 10^4\,\mathrm{m}^3\,\mathrm{kmol}^{-1}\,\mathrm{s}^{-1}$	$E_2 = 83\,680\,\mathrm{kJ}/\mathrm{kmol}$
$k_3 = 0.4167\,\mathrm{m}^3\,\mathrm{kmol}^{-1}\,\mathrm{s}^{-1}$	$E_3 = 62\,760\,\mathrm{kJ}/\mathrm{kmol}$
$\rho_1 = 639.153\,\mathrm{kg}/\mathrm{m}^3$	$\rho_2 = 607.504\,\mathrm{kg}/\mathrm{m}^3$
$\Delta H_1 = -1.04 \times 10^5\,\mathrm{kJ}/\mathrm{kmol}$	$\Delta H_2 = -1.02 \times 10^5\,\mathrm{kJ}/\mathrm{kmol}$
$\Delta H_3 = -5.5 \times 10^2\,\mathrm{kJ}/\mathrm{kmol}$	$C_p = 2.411\,\mathrm{kJ}\,\mathrm{kg}^{-1}\,\mathrm{K}^{-1}$
$Q_{1s} = -1074.63\,\mathrm{kW}$	$Q_{2s} = -6768.83\,\mathrm{kW}$
$C_p = 2.411\,\mathrm{kJ}\,\mathrm{kg}^{-1}\,\mathrm{K}^{-1}$	$R = 8.314\,\mathrm{kJ}\,\mathrm{kmol}^{-1}\,\mathrm{K}^{-1}$
Heat transfer option	*Dynamics*
Temperature approach	$77.33\,\mathrm{K}$
Heat capacity of coolant	$4.2\,\mathrm{kJ}\,\mathrm{kg}^{-1}\,\mathrm{K}^{-1}$
Medium holdup	$1000\,\mathrm{kg}$

2.4.3 First Principles Model Development

By applying the concepts of mass and energy balances, the first-principles model for the CSTRs is developed. Specifically, the dynamic model of the first CSTR is represented by the following ODEs:

$$\frac{\mathrm{d}C_{E_1}}{\mathrm{d}t} = \frac{(F_1 C_{Eo_1} - F_{out_1} C_{E_1})}{V_1} - r_{1,1} - r_{1,2} \tag{2.15a}$$

$$\frac{\mathrm{d}C_{B_1}}{\mathrm{d}t} = \frac{(F_1 C_{Bo_1} - F_{out_1} C_{B_1})}{V_1} - r_{1,1} - r_{1,3} \tag{2.15b}$$

$$\frac{\mathrm{d}C_{EB_1}}{\mathrm{d}t} = \frac{-F_{out_1} C_{EB_1}}{V_1} + r_{1,1} - r_{1,2} + 2r_{1,3} \tag{2.15c}$$

$$\frac{\mathrm{d}C_{DEB_1}}{\mathrm{d}t} = \frac{-F_{out_1} C_{DEB_1}}{V_1} + r_{1,2} - r_{1,3} \tag{2.15d}$$

$$\frac{\mathrm{d}T_1}{\mathrm{d}t} = \frac{(T_{1_o} F_1 - T_1 F_{out_1})}{V_1} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_1 C_p} r_{1,j} + \frac{Q_1}{\rho_1 C_p V_1} \tag{2.15e}$$

where $F_{out_1} = F_1$. The dynamic model of the second CSTR is represented by the following ODEs:

$$\frac{\mathrm{d}C_{E_2}}{\mathrm{d}t} = \frac{(F_2 C_{Eo_2} + F_{out_1} C_{E_1} - F_{out_2} C_{E_2})}{V_2} - r_{2,1} - r_{2,2} \tag{2.16a}$$

$$\frac{\mathrm{d}C_{B_2}}{\mathrm{d}t} = \frac{(F_2 C_{Bo_2} + F_{out_1} C_{B_1} - F_{out_2} C_{B_2})}{V_2} - r_{2,1} - r_{2,3} \tag{2.16b}$$

$$\frac{\mathrm{d}C_{EB_2}}{\mathrm{d}t} = \frac{F_{out_1} C_{EB_1} - F_{out_2} C_{EB_2}}{V_2} + r_{2,1} - r_{2,2} + 2r_{2,3} \tag{2.16c}$$

$$\frac{\mathrm{d}C_{DEB_2}}{\mathrm{d}t} = \frac{F_{out_1} C_{DEB_1} - F_{out_2} C_{DEB_2}}{V_2} + r_{2,2} - r_{2,3} \tag{2.16d}$$

$$\frac{\mathrm{d}T_2}{\mathrm{d}t} = \frac{(T_{2_o} F_2 - T_1 F_{out_1} - T_2 F_{out_2})}{V_2} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_2 C_p} r_{2,j} + \frac{Q_2}{\rho_2 C_p V_2} \tag{2.16e}$$

where $F_{out_2} = F_1 + F_2$ and the reaction rates are calculated by the following expressions:

$$r_{i,1} = k_1 \mathrm{e}^{\frac{-E_1}{RT_i}} C_{E_i} C_{B_i} \tag{2.17a}$$

$$r_{i,2} = k_2 \mathrm{e}^{\frac{-E_2}{RT_i}} C_{E_i} C_{EB_i} \quad i = 1, 2 \text{ (reactor index)} \tag{2.17b}$$

$$r_{i,3} = k_3 \mathrm{e}^{\frac{-E_3}{RT_i}} C_{DEB_i} C_{B_i} \tag{2.17c}$$

Remark 2.2. *When constructing a dynamic model based on first-principles fundamentals involving multiple ordinary differential equations (ODEs), there may be multiple potential steady states. It is crucial to design the dynamic model in a manner that ensures convergence to the desired steady state. It should be noted that the steady states obtained from the first-principles model may differ from those obtained using the Aspen model. Therefore, our approach involves expressing the first-principles dynamic model equations in the form* $\dot{x} = F(x,u) - F(x_s,u_s) = f(x) - f(x_s) + g(x)u - g(x_s)u_s$. *Here,* x_s *and* u_s *correspond to the steady-state values of the state variables and control inputs, respectively. These values are determined by the Aspen model through simulation. Writing the equations in this form guarantees that the first-principles model will converge to the desired steady states obtained from the Aspen model, particularly when dealing with multiple distinct steady states.*

2.4.4 Linking the Dynamic Models

To establish a seamless data transfer between the Aspen model (Aspen Plus Dynamics V12) and the first-principles model-based LMPC (Python code), we program a script in Aspen Plus Dynamics. This script reads the calculated control inputs, exported as text files by the Python code responsible

for computing the control inputs. Additionally, it facilitates the export of the state variable values from Aspen Plus Dynamics as text files read by the Python code. This data exchange occurs at each sampling time, establishing a robust data transfer link between the Aspen model and the first-principles-based LMPC.

Remark 2.3. *As discussed in Section* 2.4.1*, the MPC model (first-principles based) used for predicting future states and optimizing control inputs differs from the Aspen dynamic model, where we apply the controller. To address this model mismatch, we analyze the combined and relative effects of quantization errors, which arise from encryption-decryption and can further amplify the model mismatch error. Our analysis reveals that the quantization error is bounded by half the resolution (resolution/2). For instance, when the quantization parameter chosen is* $d = 1$*, the resolution is 0.5, and the upper bound of the error between the actual and quantized values is resolution/2 or 0.25. Hence, for higher quantization parameters, the impact of quantization error on the overall model mismatch error is negligible. It is important to note that quantization introduces a bounded error in the states, thereby limiting the extent of the model mismatch error.*

2.4.5 Implementing the encrypted LMPC

Before implementing encryption-decryption in a process, it is crucial to carefully choose the values: d_1, l_1, and l_2. After closely examining the maximum and minimum permissible values of the states and inputs, we determine the number of integer bits, $l_1 - d_1$. The largest value in the set $\mathbb{Q}_{l_1,d}$ is obtained using the formula $2^{l_1-d_1-1} - 2^{-d_1}$, while the smallest value is $-2^{l_1-d_1-1}$. The quantization parameter d_1 should be selected based on factors such as desired accuracy and the range of state and input values. Additionally, a value for l_2 should be selected such that l_2 is greater than l_1.

These steps complete the hyperparameter selection process.

After following the aforementioned steps, we determine that, for the example discussed in this section, $l_1 - d$ is calculated to be 15. The values of l_1 and d need to be selected accordingly. In the set $\mathbb{Q}_{l_1,d}$, rational numbers are separated by a resolution of 2^{-d}, meaning that a higher value of d leads to lower quantization errors. For simulation purposes, we vary the values of d from 1 to 8, resulting in l_1 ranging from 16 to 23. It is important to ensure that $l_2 > l_1$ for the bijective mapping, so we choose $l_2 = 30$. After determining all the quantization related parameters, we proceed to quantize the states and inputs. Subsequently, we encrypt them using the Paillier Encryption algorithm. The implementation of Paillier Encryption is carried out using the "phe" module in Python, specifically PythonPaillier [21]. The first-principles model, described by equations Eq. (2.15) and Eq. (2.16), serves as the process model in the LMPC framework. To solve the optimization problem, we utilize the Python module of the IPOPT software [83].

Remark 2.4. *IPOPT, Interior Point OPTimizer, is a software tool designed specifically for solving nonlinear optimization problems. It employs an iterative method known as the interior point method, which focuses on finding the optimal solution by gradually moving towards the interior of the feasible region. To solve the optimization problem, IPOPT employs a series of iterations. In each iteration, it updates a sequence of points that satisfy the given constraints and improve the value of the objective function. This process involves calculating descent directions based on the gradient and Hessian of both the objective function and the constraints. IPOPT considers both the feasibility and optimality of the solution, striving to find a point that not only satisfies the constraints but also optimizes the objective function. Throughout the iterations, IPOPT uti-*

lizes a barrier function to handle inequality constraints and a penalty function to handle equality constraints. It also incorporates a line search procedure to determine the appropriate step length and employs backtracking techniques to ensure convergence towards the optimal solution. In our study, the nature of the MPC formulation leads to a non-convex optimization problem. This signifies that the optimum achieved through the IPOPT optimizer is a local optimum, rather than a global one. The optimization process begins with a designated starting input trajectory based on predicted values for the extended horizon (beyond the first input trajectory calculation) from the prior iteration. Furthermore, the optimizer is guided by a prescribed tolerance error and an upper limit on the number of iterations. The optimizer will persist in its pursuit of an improved solution until either of these conditions is met. If the optimizer is unable to calculate an optimal solution, the computed solution from the backup controller (P-controller) will be substituted for that specific sampling instance.

To implement encryption in a practical setting, it is crucial to ensure that the sampling time, Δ exceeds the combined maximum of the encryption-decryption time required for all the states and control inputs, as well as the maximum time needed for computing the control action at each sampling instance for all the considered quantization parameters, denoted as d. This requirement can be expressed by the following equation:

$$\begin{aligned} &\Delta > \max(\text{Enc-Dec time}) + \max(\text{MPC computation time}) \\ &\forall\, d = \{1, 2, 3, 4, 5, 6, 7, 8\} \end{aligned} \tag{2.18}$$

During the implementation of the encrypted MPC design in SCADA systems, where encrypted sensor measurements and control actions are transmitted through the network, the time

spent on signal transmission is generally not substantial due to the rapid and efficient nature of networked communication. However, this efficiency comes at the risk of susceptibility to cyberattacks. To mitigate this potential vulnerability, this study encrypts these communication channels and assesses the repercussions of encryption. Consequently, the formula provided above does not incorporate the factor of signal transmission time as well as issues with asynchronous, delayed measurements that have been studied in past works [54, 55]. The sampling time, Δ is carefully selected as 30 seconds, considering the aforementioned condition to ensure proper implementation. The integration step h_c is chosen as $(10^{-2}$ x $\Delta)$ to evaluate the cost function of the LMPC through the first-principles model. The positive definite matrix P in the control Lyapunov function $V = x^T P x$ for this system is taken as $P = \text{diag}[200\ \ 500\ \ 2500\ \ 10\ \ 0.25\ \ 1000\ \ 1000\ \ 500\ \ 1\ \ 0.5]$ based on extensive simulations. A prediction horizon of $N = 6$ is employed in the LMPC framework. To ensure stability in the LMPC, we set the criterion $\rho_{min} = 2$ to determine when the states have reached stability. Additionally, a contractive constraint of the form $\dot{V} \leq -kV$ is utilized for Eq. (2.13f), where the value of k is chosen as 0.15. The weight matrices Q_1 and Q_2 in the LMPC cost function are chosen as $Q_1 = \text{diag}[5\ \ 5\ \ 650\ \ 5\ \ 2.5\ \ 25\ \ 25\ \ 100\ \ 2\ \ 6]$ and $Q_2 = \text{diag}[5$ x $10^{-6}\ \ 1.25$ x $10^{-5}]$, respectively. The cost function is defined as $L_2(x(t), u(t)) = x^T Q_1 x + u^T Q_2 u$.

2.4.6 Utilizing MPC over Traditional Control

In this section, we substantiate the utilization of model predictive control (MPC) by conducting a comparative analysis between the MPC and the simpler p-control strategy. P-control allows control actions to be computed directly in encrypted states, eliminating the requirement for decryption at

the controller through complex multiplicative homomorphic algorithms such as the ElGamal cryptosystem. The MPC strategy is a more advanced control method that uses a mathematical model of the system to predict future behavior and optimize control actions accordingly. It requires decryption at the controller to obtain the necessary information for prediction and multi-constrained, non-linear optimization, which cannot be performed in an encrypted space.

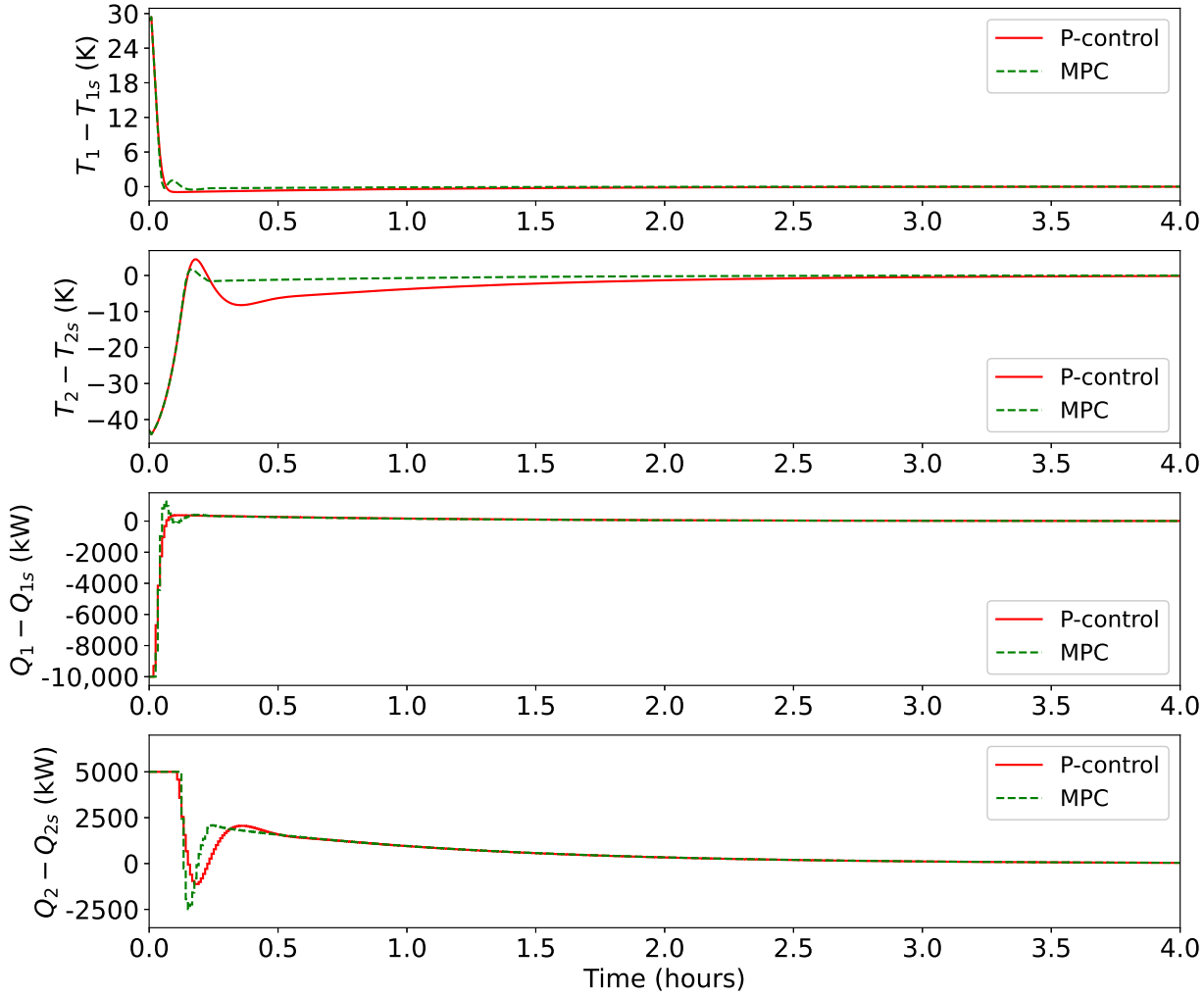

Figure 2.4: Temperature state and input profiles of P-control (red solid line) and MPC (green dashed line) strategies employed using the Aspen dynamic model.

Figure 2.4 showcases its enhanced performance, with lower undershoot and faster settling

time observed for the temperature of CSTR 1. Further, the temperature of CSTR 2 exhibits a significant reduction in overshoot by almost 50% and converges over 1 hour before the p-control, within a settling limit of $0.25\,\mathrm{K}$. Moreover, the evaluation of the normalized sum of the controller cost function ($L_2(x(t), u(t))$) over the closed-loop simulation, reinforces the advantage of MPC over p-control, by the respective values of 0.86 and 1. These findings underscore the necessity of adopting MPC, as it offers reduced overshoot, undershoot, faster settling time of state variables, and enhanced cost efficiency.

Remark 2.5. *As mentioned earlier in Section* 2.2.4*, the Paillier cryptosystem is a partially homomorphic encryption scheme that does not support multiplication operations in an encrypted space. Therefore, in the above section, we mention using the ElGamal Cryptosystem, which supports multiplicative homomorphism. Although the Paillier cryptosystem supports addition operations in encrypted space, we do not utilize this property in our study. The Paillier cryptosystem is primarily selected for encryption due to its lower computational complexity compared to the ElGamal cryptosystem. This choice reduces the time and computational effort required for encryption-decryption processes.*

2.4.7 Simulation results of the Encrypted LMPC

We apply the encrypted LMPC to the Aspen dynamic model, initialized from the point:

$x_0 = [-1.11\,\mathrm{kmol/m^3} \;\; -1.16\,\mathrm{kmol/m^3} \;\; -0.3\,\mathrm{kmol/m^3} \;\; -8.76 \text{ x } 10^{-6} kmol/m^3 \;\; 28.85\,\mathrm{K}$

$0.49\,\mathrm{kmol/m^3} \;\; 0.56\,\mathrm{kmol/m^3} \;\; -1.85\,\mathrm{kmol/m^3} \;\; -7.77 \text{ x } 10^{-6} kmol/m^3 \;\; -43\,\mathrm{K}]$

We then observe the closed-loop simulation results for $d = 1, 4, 8$. A process time of 4 hours allows both the states and control inputs to reach their respective steady-state values. Figures 2.5

to 2.7 display the temperature state and input profiles.

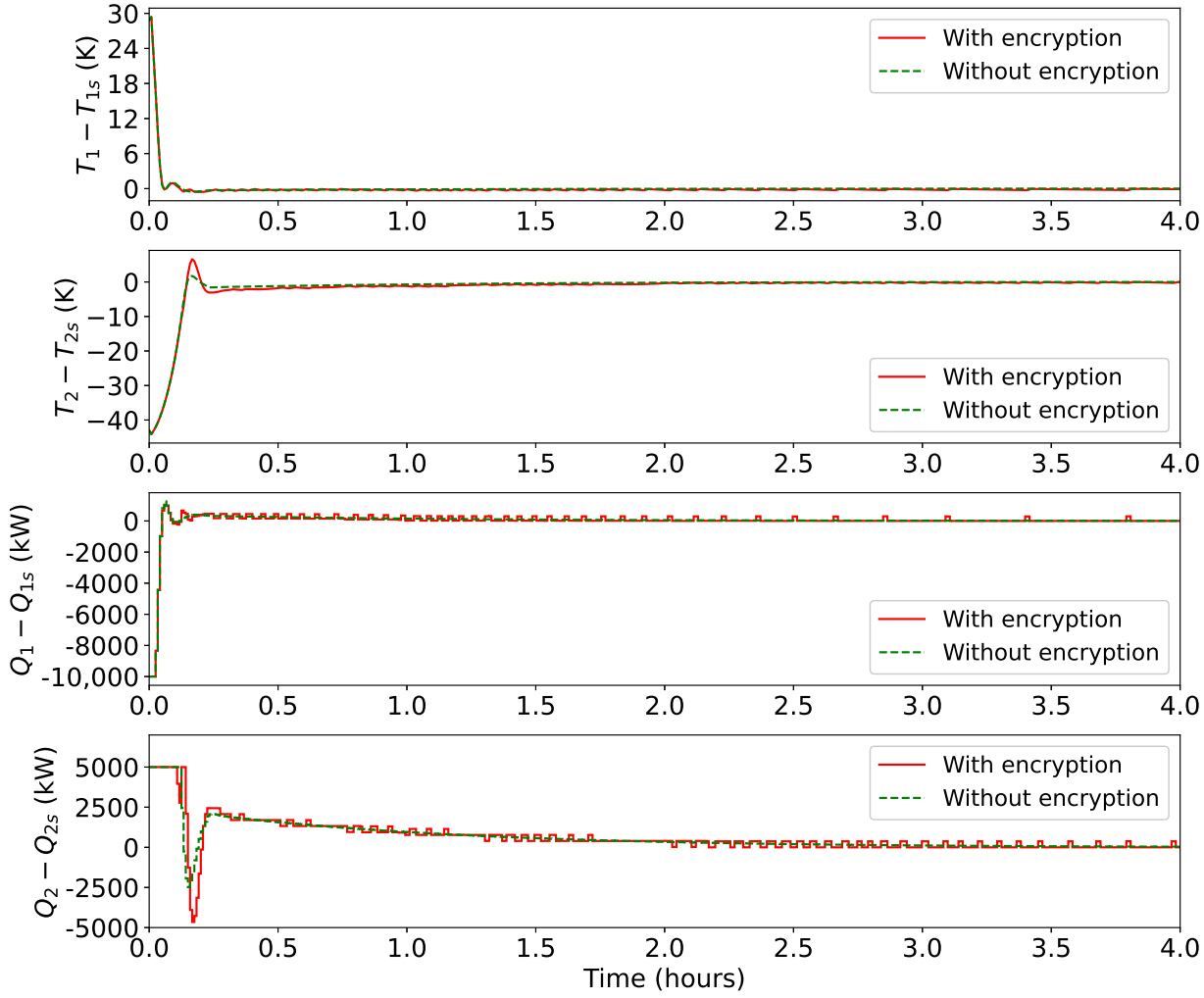

Figure 2.5: Temperature state and input profiles of the LMPC with encryption (red solid line) and without encryption (green dashed line) for the Aspen dynamic model, with $d = 1$.

Remark 2.6. *As indicated in Section* 2.4.1*, the concentration states in the reactors exhibit open-loop trajectories as the reactor temperature converges to its steady-state value. Consequently, the presence or absence of encryption does not significantly affect these states since the manipulated input, i.e., the heat removed from the reactors, has no direct influence on the concentration states. Therefore, in this section, we focus solely on displaying the temperature states and control inputs,*

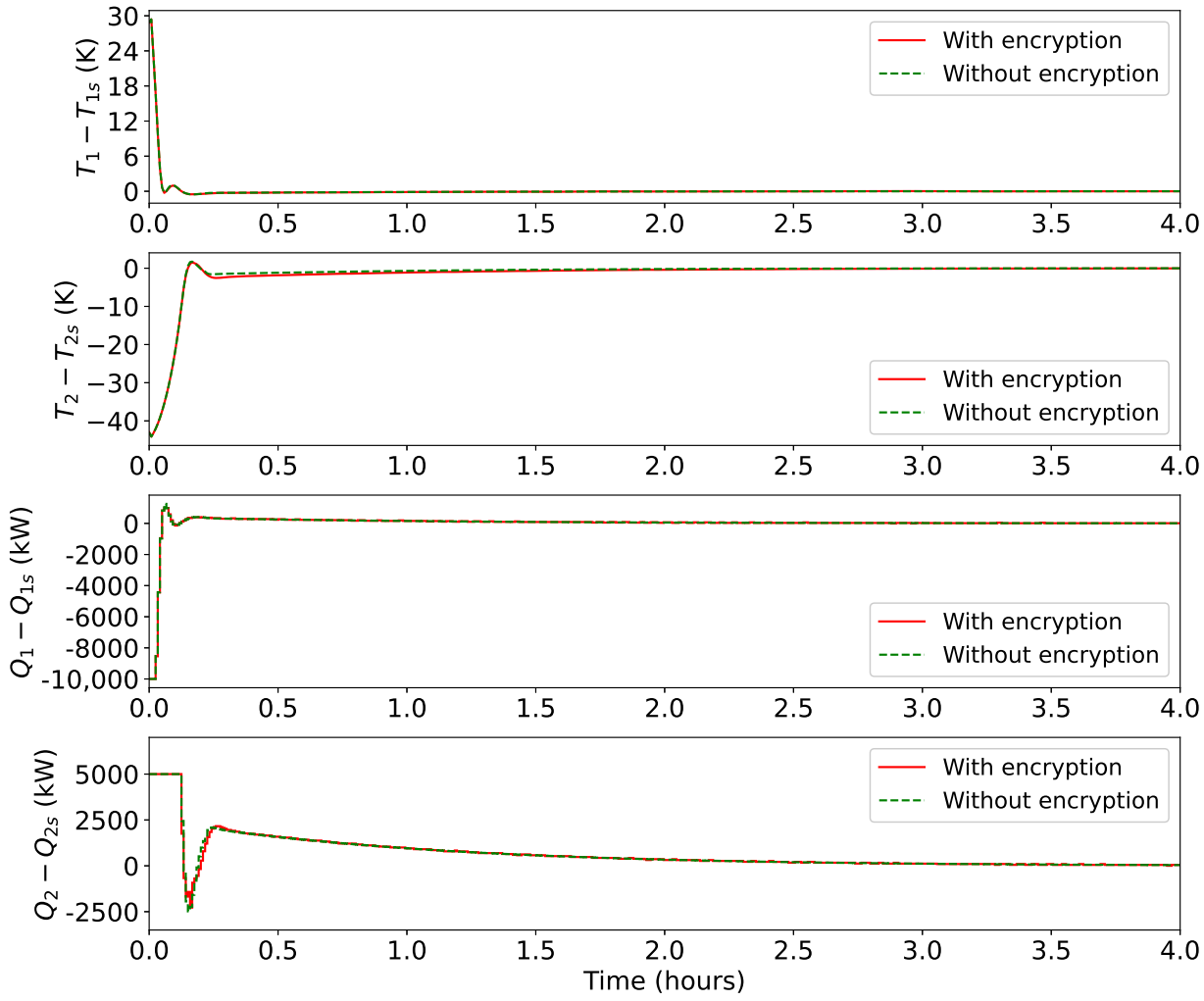

Figure 2.6: Temperature state and input profiles of the LMPC with encryption (red solid line) and without encryption (green dashed line) for the Aspen dynamic model, with $d = 4$.

as encryption noticeably influences them.

For a quantization parameter of $d = 1$, it is evident that the state $T_1 - T_{1s}$ does not precisely converge to its steady-state value, instead exhibiting small oscillations around it throughout the 4-hour process time. Also, the state $T_2 - T_{2s}$ demonstrates nearly double the overshoot with encryption and oscillates around the steady-state values, similar to the previous state. Further, quantized control inputs $Q_1 - Q_{1s}$ and $Q_2 - Q_{2s}$ experience significant oscillations under the

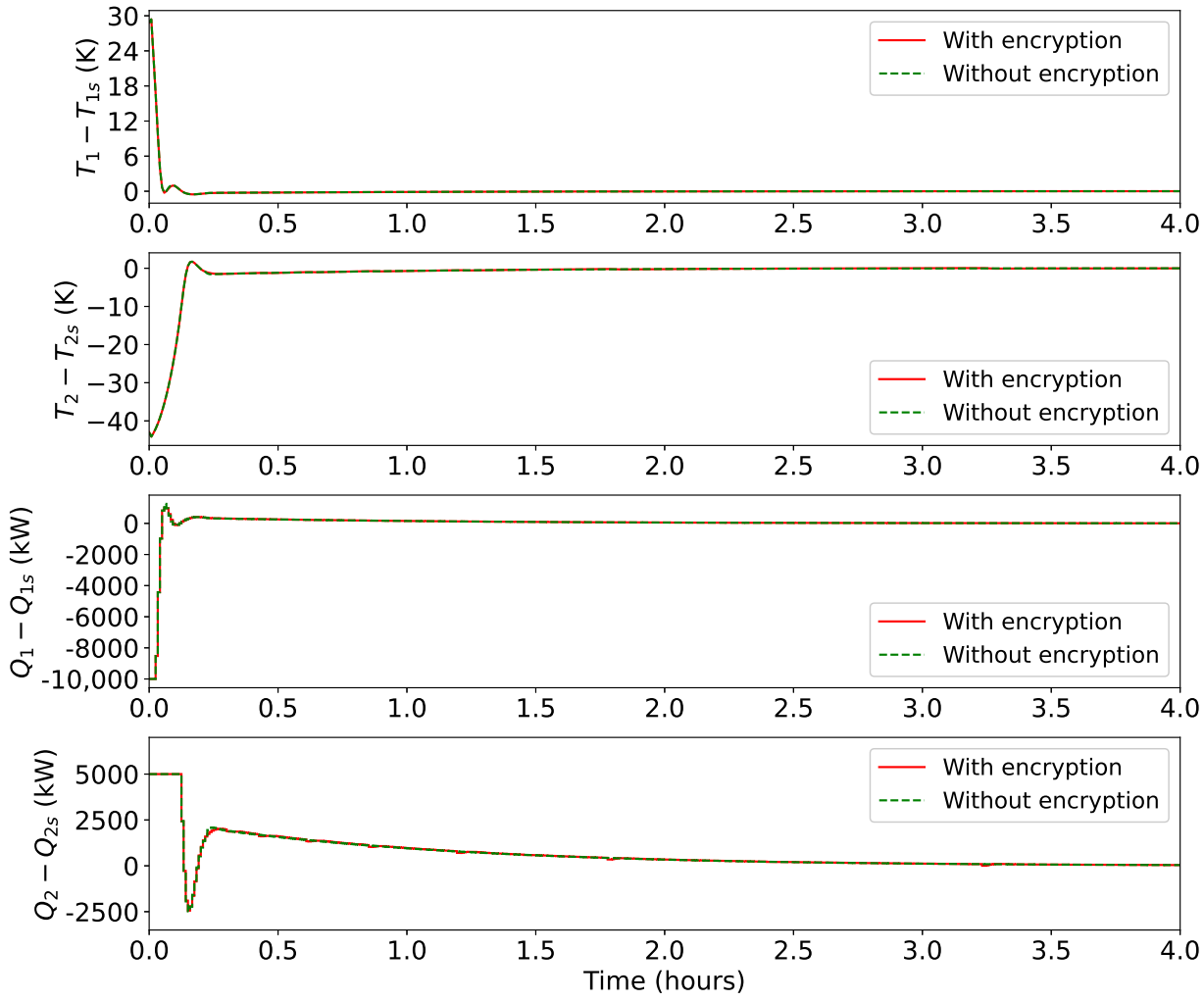

Figure 2.7: Temperature state and input profiles of the LMPC with encryption (red solid line) and without encryption (green dashed line) for the Aspen dynamic model, with $d = 8$.

encrypted MPC, rendering it incapable of effectively stabilizing the closed-loop system within a small neighborhood $\Omega_{\rho_{\min}}$ around the origin. Although, it does stabilize the system within the larger neighborhood $\Omega_{\hat{\rho}}$. This behavior can be attributed to the quantization error resulting from the quantization of the state measurements. Thus, we establish that errors due to quantization can be more significant than plant/model mismatch errors as the MPC without encryption and with a higher quantization parameter, $d = 8$, is stabilized within the small neighborhood $\Omega_{\rho_{\min}}$ around the

origin. As indicated in Remark 2.7, the quantization error associated with the quantized control input can be deemed negligible. However, the quantization error emanating from the quantized states is significant given the range in which they lie during closed-loop simulation. For $d = 1$, the quantized states are separated by a resolution of 2^{-1} or 0.5, leading to a high quantization error. When running simulations with the quantization parameter $d = 4$, we no longer observe oscillatory motions in the temperature states, and the magnitude of oscillations for the quantized inputs is much smaller compared to the case where $d = 1$. Furthermore, the amplitude of overshoot observed in the state variable $T_2 - T_{2s}$ remains nearly unchanged when encryption is applied, and the system also reaches the steady-state more rapidly.

It is important to note that as the quantization parameter increases, resulting in a lower resolution, the states and inputs converge more quickly and exhibit reduced oscillations. Therefore, a higher quantization parameter improves the convergence behavior and decreases fluctuations in the state and control input profiles. Specifically, when $d = 8$, the closed-loop trajectories of the temperature states and control inputs become nearly identical between the cases with encryption and without encryption. In other words, the impact of encryption on the system's behavior diminishes significantly as the quantization parameter increases, ultimately resulting in almost indistinguishable closed-loop trajectories for both scenarios.

Remark 2.7. *The total quantization error can be attributed to the state quantization rather than the control input quantization, since the magnitudes of the quantized control inputs generally fall within the order of magnitude three. For the case* $d = 1$*, representing the lowest quantization, the maximum permissible error in the control input calculated by the MPC (before encryption) and*

applied by the actuator (after decryption) is 0.25, corresponding to half of the resolution. This error is considered negligible compared to the overall control input. As a result, the error arising from the quantization of control inputs is insignificant, particularly for the specific example considered. However, it is crucial to acknowledge that if the control inputs have smaller magnitudes, the error resulting from the quantized control inputs would significantly impact the controller's performance.

2.5 Effect of the quantization parameter *d* and Encryption-Decryption on the total computational time

This section discusses the impact of the quantization parameter, d, and encryption-decryption on the total control input calculation time. For an encrypted MPC, the total control input calculation time comprises two main components: the time required by the MPC to calculate the control action and the total time spent on encrypting-decrypting the state variables and control inputs.

2.5.1 Effect of the quantization parameter *d* on computational time

Table 2.2 provides an overview of the computation time required for the complete encryption-decryption process, considering a range of quantization parameters, $d = [1, 8]$. The table also offers a detailed breakdown of the time required for each sub-process involved. Analyzing Table 2.2, it becomes apparent that the computational time for the entire encryption-decryption process shows consistent values across the quantization parameters within the range $d = [1, 8]$.

However, as discussed in Section 2.4, a higher quantization parameter proves more advanta-

Table 2.2: Time required to encrypt-decrypt the 10 states and 2 inputs at a single sampling instance

d	$g_{l_1,d}$ Time	$f_{l_2,d}$ Time	Enc. Time	Dec. Time	$f_{l_2,d}^{-1}$ Time	Total Time
1	$4.8 \times 10^{-4}\,\mathrm{s}$	$2.6 \times 10^{-4}\,\mathrm{s}$	$2.49\,\mathrm{s}$	$0.72\,\mathrm{s}$	$2.9 \times 10^{-4}\,\mathrm{s}$	$3.204\,\mathrm{s}$
2	$4.5 \times 10^{-4}\,\mathrm{s}$	$2.9 \times 10^{-4}\,\mathrm{s}$	$2.48\,\mathrm{s}$	$0.71\,\mathrm{s}$	$3.1 \times 10^{-4}\,\mathrm{s}$	$3.190\,\mathrm{s}$
3	$4.7 \times 10^{-4}\,\mathrm{s}$	$2.7 \times 10^{-4}\,\mathrm{s}$	$2.48\,\mathrm{s}$	$0.7\,\mathrm{s}$	$2.8 \times 10^{-4}\,\mathrm{s}$	$3.179\,\mathrm{s}$
4	$4.8 \times 10^{-4}\,\mathrm{s}$	$2.9 \times 10^{-4}\,\mathrm{s}$	$2.48\,\mathrm{s}$	$0.71\,\mathrm{s}$	$2.8 \times 10^{-4}\,\mathrm{s}$	$3.182\,\mathrm{s}$
5	$5.3 \times 10^{-4}\,\mathrm{s}$	$2.7 \times 10^{-4}\,\mathrm{s}$	$2.5\,\mathrm{s}$	$0.71\,\mathrm{s}$	$2.8 \times 10^{-4}\,\mathrm{s}$	$3.214\,\mathrm{s}$
6	$5 \times 10^{-4}\,\mathrm{s}$	$2.9 \times 10^{-4}\,\mathrm{s}$	$2.47\,\mathrm{s}$	$0.71\,\mathrm{s}$	$3.2 \times 10^{-4}\,\mathrm{s}$	$3.182\,\mathrm{s}$
7	$5.1 \times 10^{-4}\,\mathrm{s}$	$3 \times 10^{-4}\,\mathrm{s}$	$2.49\,\mathrm{s}$	$0.71\,\mathrm{s}$	$3.3 \times 10^{-4}\,\mathrm{s}$	$3.194\,\mathrm{s}$
8	$5.4 \times 10^{-4}\,\mathrm{s}$	$2.9 \times 10^{-4}\,\mathrm{s}$	$2.5\,\mathrm{s}$	$0.73\,\mathrm{s}$	$3.1 \times 10^{-4}\,\mathrm{s}$	$3.225\,\mathrm{s}$

geous for the LMPC. Specifically, for $d = 8$, the trajectories of the temperature states and control inputs closely resemble those without encryption. In contrast, for $d = 1$, there is a noticeable difference between the cases with and without encryption.

Furthermore, this table also reveals that the majority of the computational time is allocated to the encryption step, followed by the decryption step. Mapping the inputs to quantized states ($g_{l_1,d}$), bijective mapping ($f_{l_2,d}$), and inverse mapping ($f_{l_2,d}^{-1}$) contribute only a negligible fraction of the total time at each sampling instance. Although the computational time remains consistent across quantization parameters, the number of search operations at each sampling instance increases linearly with the quantization parameter. This observation is presented in Table 2.3. Additionally, the time and number of operations required to generate the set $\mathbb{Q}_{l_1,d}$ grow exponentially by increasing the quantization parameter, d. However, it is vital to note that this step is performed only once at the beginning of the process and is not repeated at each sampling instance. Consequently, selecting a higher quantization parameter remains favorable, as the operational time for encryption-decryption at each sampling instance remains unchanged, and a higher quantization parameter yields significantly improved results.

Table 2.3: Operations required for $g_{l_1,d}$, generating $\mathbb{Q}_{l_1,d}$, and time required to generate $\mathbb{Q}_{l_1,d}$

d	Operations for $g_{l_1,d}$	Operations to generate $\mathbb{Q}_{l_1,d}$	Time to generate $\mathbb{Q}_{l_1,d}$
1	192	65534	$0.02\,\mathrm{s}$
2	204	131070	$0.04\,\mathrm{s}$
3	216	262142	$0.07\,\mathrm{s}$
4	228	524286	$0.15\,\mathrm{s}$
5	240	1048574	$0.29\,\mathrm{s}$
6	252	2097150	$0.55\,\mathrm{s}$
7	264	4194302	$1.11\,\mathrm{s}$
8	276	8388606	$2.27\,\mathrm{s}$

Remark 2.8. *An alternative approach to mitigate the initial high computational time, especially when a higher quantization parameter d is selected, is to pre-generate the set $\mathbb{Q}_{l_1,d}$ before commencing the process operation with encryption-decryption on the hardware. By generating this set prior to the first sampling instance, we can avoid the need for additional time allocation during the actual control process. As mentioned in, Section* 2.5 *as the quantization parameter increases, the time required to generate $\mathbb{Q}_{l_1,d}$ grows exponentially. Therefore, pre-generating the set is particularly beneficial in reducing computational overhead during the initial sampling instance when dealing with larger quantization parameters. This approach allows for the utilization of higher quantization parameters without being hindered by the drawback of increased computational time in the first sampling instance.*

2.5.2 Effect of Encryption-decryption on the total computational time

The Figure 2.8 shows that encryption-decryption takes approximately 45-65% of the total time required to calculate the control inputs for an encrypted LMPC, which is the sum of the time needed for MPC control action computation and encryption-decryption (of the 10 states and 2 control inputs). Moreover, this result is consistent over the quantization parameters $d = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

This substantiates the fact that the decision regarding the choice of a quantization parameter does not necessarily result in a substantial alteration of the ratio between the time devoted to encryption-decryption and the total duration of MPC computation.

As previously discussed in Section 2.4.5, it is essential to select a sampling time, Δ, that exceeds the combined maximum duration of the encryption-decryption process and the MPC computation time for any given sampling instance. This criterion applies to all considered quantization parameters. For the example examined in this study, the minimum required sampling time was determined to be $9\,\mathrm{s}$. Consequently, a sampling time of $30\,\mathrm{s}$ was selected, which exceeded the minimum requirement.

Remark 2.9. *Maintaining system stability and ensuring effective control requires avoiding excessively large sampling times, particularly in cases where the system operates at an unstable steady-state and has bounded control inputs. Going beyond a certain threshold in sampling time can impede the ability of the controller to successfully regulate the system. To validate this concept, we conducted an experiment on the example discussed in Section* 2.4. *We applied LMPC control without encryption and increased the sampling time for the process in 30-second increments. The results showed that the controller achieved the desired steady-state for a sampling time up to 2.5 minutes. However, extending the sampling time to 3 minutes prevented the controller from achieving the desired outcome. This observation emphasizes the significance of selecting an appropriate sampling time that ensures effective control action and system stability.*

Remark 2.10. *In order to maintain manageable encryption-decryption times within an encrypted control system network, it is essential to choose computationally efficient cryptosystems, such as*

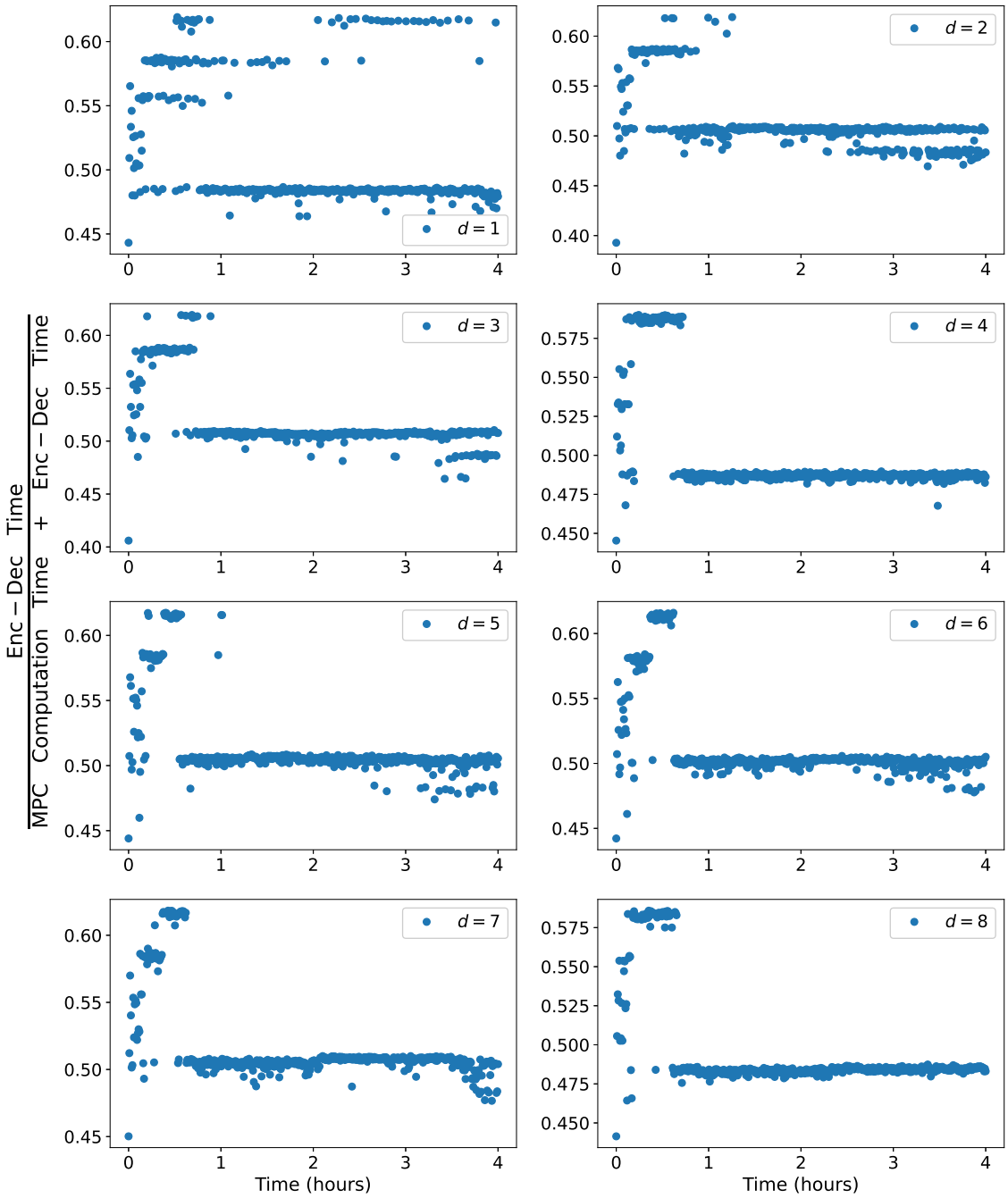

Figure 2.8: Ratio of the total time spent for encryption-decryption to the sum of the total time required for MPC computation and encryption-decryption at each sampling instance.

the Paillier Cryptosystem. Cryptosystems like ElGamal and AES impose higher computational requirements on process control hardware, resulting in longer encryption and decryption times.

Consequently, this leads to the need for longer sampling times. Further, in practical applications, it may be feasible to reduce the prediction horizon of the MPC for encrypted control as long as it does not significantly impact the performance of the controller. These adjustments enable shorter sampling times while still meeting encryption requirements.

Remark 2.11. *When dealing with large-scale processes with hundreds or thousands of measurements, it would be advisable to employ a distributed SCADA architecture across multiple locations or nodes within the network. Furthermore, encryption of state measurements at the sensor can be performed in parallel rather than in series. When we report the encryption time in this paper, it is the total time needed for encrypting each sensor signal and control input in series, not in parallel. This could be done in a parallel manner across multiple devices for larger systems to reduce the effective computational time needed.*

Remark 2.12. *To deal with asynchronous or delayed signals in an encrypted setting, the signals would be encrypted prior to transmission and decrypted upon receipt, with the actuator designed to apply control inputs in a sample-and-hold manner, whereby the preceding control input trajectory continues to be implemented until the recalculated input trajectory is received. Since quantization with encryption has consistent computational duration, an appropriate sampling time would be chosen based on its knowledge and time needed to compute the control input, as demonstrated in Eq.* (2.18). *However, because the formula given to decide the sampling time does not take into account time spent for signal communication or signal delays, which are very specific to the process setting, sensors used, and communication channels established, the time spent between asynchronous measurements or for signal delays could be known or approximated to select an*

appropriate sampling time.

2.6 Conclusions

In this chapter, we developed and applied an Encrypted Lyapunov-based model predictive control (LMPC) Scheme to a large-scale chemical process network involved in the production of ethylbenzene. By employing the encrypted LMPC, we conducted closed-loop simulations for different quantization parameters and identified errors resulting from quantization. We illustrated that the effect of quantization could be more profound than plant/model mismatch when a low quantization parameter is chosen. To mitigate the impact of quantization, we proposed using a higher quantization parameter, specifically $d = 8$. Furthermore, through a comprehensive analysis of the duration of encryption-decryption at each sampling instance, we observed that the computational burden on the control input calculation time remained consistent across all tested quantization parameters. This finding supports the recommendation of employing a higher quantization parameter, as it not only minimizes the impact of quantization errors but also ensures secure communication between the sensor-controller and controller-actuator, thus enhancing system cybersecurity without compromising the performance of the controller. The current research necessitates MPC computations to be executed within a fully secure cyber-physical environment, aimed to thwart cyberattackers from compromising the decrypted plaintext input signals and control inputs computed by the MPC prior to encryption. An avenue for future research could involve adapting the encrypted MPC architecture to operate within a less secure context. Additionally, another promising area for future investigation could entail implementing encrypted MPC while incorporating data reconciliation

mechanisms amidst a cyberattack scenario. Notably, the works referenced [58, 75] in this context have explored such aspects within non-encrypted settings.

Chapter 3

Integrating machine learning detection and encrypted control for enhanced cybersecurity of nonlinear processes

3.1 Introduction

The swift advancements in technology and the increasing integration of devices have made interconnected cyber-physical systems essential elements of vital infrastructure in various sectors like energy, water, transportation, and manufacturing. In particular, systems that employ SCADA (Supervisory Control and Data Acquisition) technology play a crucial role in overseeing, directing, and automating intricate operations, thereby boosting efficiency and productivity. Nonetheless, the expanded interlinking and fusion of SCADA systems with the internet and corporate networks have made them susceptible to potential cyber threats. A breach or compromise within these systems could lead to grave outcomes, including disruption of essential services, physical harm, financial

setbacks, and even jeopardizing public safety. Current advancements in cyberattack methodologies underscore the importance of instituting robust cybersecurity measures.

While notable strides have been made in tackling cybersecurity issues within the domain of information technology (IT), the operational technology (OT) domain is currently lagging behind in terms of advancements. IT predominantly concentrates on the software aspect of systems, covering areas like network architecture and data administration. On the other hand, OT is responsible for maintaining the seamless functioning of essential infrastructure, such as power grids, intelligent meters, and distribution networks. Cyberattacks on OT infrastructure can result in consequences such as operational shutdowns, service disruptions, data leaks, and potentially catastrophic explosions. As an illustration, consider the case of the Stuxnet malware, which was uncovered in 2010. This particular malicious software was designed with a specific focus on infiltrating SCADA systems. Stuxnet managed to breach programmable logic controllers (PLCs) within Iranian nuclear facilities, collecting valuable information about the industrial system and ultimately causing the high-speed centrifuges to burnout [47]. Another noteworthy incident involves the cyberattack on the Ukrainian power grid in 2015. During this event, hackers infiltrated SCADA systems to remotely shutdown substations, resulting in power failures. A more recent occurrence took place in 2021, concerning the Colonial Pipeline, a major operator of fuel pipelines in the United States. This company fell victim to a ransomware attack, orchestrated by hackers who gained entry through the use of the DarkSide ransomware. The attackers proceeded to encrypt the networked communication of the pipeline, demanding a ransom payment in return for the decryption keys. Consequently, Colonial Pipeline had to cease its operations, resulting in interruptions to fuel distribution and causing notable financial loss. These examples underscore the imperative for robust cybersecurity

protocols in OT infrastructures.

Extensive research efforts continue to focus on various domains, such as the design of backup controllers in a two-tier safety-performance control architecture [13], the creation of machine learning-based cyberattack detectors [1, 36, 66, 84], the recovery of process states following a cyberattack [87], the development of cyberattack-resilient controllers [24, 25], and encrypted control [81]. However, this research aims to integrate some of these approaches, particularly machine-learning based cyberattack detection in a two-tier encrypted control architecture, to create a robust and cyber-secure control scheme applicable to nonlinear processes.

Networked communication lines are vulnerable to cyberattacks when data is transmitted in its regular plaintext form. To address this, encryption emerges as a solution, effectively safeguarding data during its transfer. Within control systems, data serves as the foundation for computing control inputs. While encryption offers enhanced security, it also introduces limitations, allowing only linear computations—a drawback that can hamper the utilization of advanced controllers like model predictive control (MPC) in complex systems characterized by nonlinear dynamics.

MPC ensures closed-loop stability (confinement of system states within a level set of the control Lyapunov function), optimizes critical performance metrics, handles multi-input multi-output scenarios, and manages constraints on system states and inputs. These advantages stem from the deployment of a mathematical model to predict future behavior and consequently optimize control inputs by minimizing a cost function. However, for the application of MPC, decryption becomes necessary to provide the required measurements for prediction and optimization at the end of the controller. While a linear control law provides the ability to calculate control inputs in an encrypted space, eliminating the need for decryption and ensuring a more secure approach, the advantages of

nonlinear model predictive control cannot be ignored. Moreover, a delicate balance exists between improving system cybersecurity and enhancing closed-loop performance. Thoughtful assessments are necessary, taking into account the improvement achieved with the nonlinear controller, the level of cybersecurity in the process setting, and, most crucially, the adherence to the necessary physical safety standards for the process. Similarly, the selection of a nonlinear controller, even with the aim of improving closed-loop control performance, might not be justified if it increases the vulnerability of the system to cyberthreats.

To reconcile the benefits of both paradigms, we propose an encrypted two-tier control architecture coupled with ML-based cyberattack detection. In this setup, the lower tier is composed of an encrypted linear control scheme capable of calculating control inputs within an encrypted space, eliminating the requirement for decryption in the network. This self-contained lower tier is capable of independently stabilizing the system. Conversely, the upper tier comprises an encrypted nonlinear controller (e.g., MPC) that receives encrypted signals which are decrypted to plaintext upon arrival to compute control inputs. The computed plaintext control inputs are subsequently encrypted before transmission to the actuator. It is crucial to emphasize that the plaintext data received by the MPC and the computed plaintext control inputs are both susceptible to cyberattacks in the networked upper tier.

However, with ML-based cyberattack detection integrated in the encrypted control architecture, when a cyberattack is detected, the compromised upper tier is deactivated, and exclusively the secure and stabilizing lower tier is utilized to regain system stability. This approach enables us to amalgamate the strengths of cyber-secure encrypted linear control and advanced nonlinear control, to create a cyber-secure, advanced nonlinear control scheme that fortifies the system against

cyberattacks. Beyond ML-based cyberattack detection, alternative detection strategies can be considered. These include a reachable set-based detection scheme as explored in the work of [63], where a set is created that includes all possible states that a system can reach or achieve under specific control inputs and initial conditions. Deviations from these expected states could indicate a potential cyberattack. However, this method is restricted to linear systems. Another approach involves employing a controller switching technique, wherein controller-observer parameter switching occurs between nominal system parameters and attack-sensitive system parameters to facilitate attack detection [61, 62]. However, this method may fail to detect intelligent cyberattacks which are designed to avoid detection by conventional metrics such as residual errors. However, this study only focuses on intelligent cyberattacks, which are discussed in Section 3.4.

In the previous work of [81], it was assumed that the computing unit responsible for decrypting states and computing control inputs is cybersecure. However, in this current study, we have developed a more robust control framework. Even if the computing unit is not secure and comes under a cyberattack, our control system logic deactivates the upper-tier controller and solely relies on the encrypted lower-tier controller to stabilize the system. This lower-tier control is linear and operates within an encrypted space and does not share access to public and private keys with the computing unit, unlike the upper-tier controller, which is nonlinear. Consequently, even in scenarios where the environment for computing control inputs is not considered cybersecure, our proposed control framework can be used to enhance cybersecurity. As an alternative to a secure encrypted lower tier, a locally secure tier with backup sensors could potentially be employed [13]. However, employing an encrypted lower tier ensures a continuous and seamless flow of encrypted network communication, which can solely be accessed by authorized personnel equipped with the

required private keys necessary for decryption. Consequently, this approach eliminates the necessity for secure local communication that is isolated from the network, which poses challenges in terms of access. This distinctive aspect underscores the novelty and significance of this research.

The subsequent sections of this paper are structured as follows: in Section 3.2, we present the notation, describe the class of systems employed, explain the cryptosystem applied for encryption, and the implications of quantization; In Section 3.3 we elaborate on the architecture design of the encrypted two-tier control, outline the formulation of both the encrypted lower tier and upper tier, followed by a stability analysis to identify sources of errors in the control framework and set bounds to it; In Section 3.4 we describe the various launched cyberattacks and the machine-learning-based cyberattack detector; in Section 3.5, we showcase the application of the proposed control scheme on a nonlinear chemical process network, explain the important points to be considered while implementing the control framework in nonlinear systems, and put forth the computational load arising from the incorporation of ML-based detection within the encrypted control scheme.

3.2 Preliminaries

3.2.1 Notation

The symbol $\|\cdot\|$ represents the Euclidean norm of a vector. The transpose of the vector x is denoted by $x^\top$. $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ denote the sets of real numbers, integers, and natural numbers, respectively. Moreover, the notations $\mathbb{Z}_M$ and $\mathbb{Z}_M^*$ are used to represent the additive and multiplicative groups of integers modulo M, correspondingly. The operation of subtracting sets is indicated by the symbol "\", such that $A\backslash B$ denotes the set of elements present in A but not in B. A function denoted

as $f(\cdot)$ is categorized as belonging to class $\mathcal{C}^1$ if it possesses continuous differentiability within its domain. A function $\alpha : [0, a) \rightarrow [0, \infty)$ is categorized within the class $\mathcal{K}$ when it is strictly increasing and $\alpha(0) = 0$. The term $\mathrm{lcm}(i, j)$ indicates the least common multiple of the integers i and j. The term $\gcd(i, j)$ indicates the greatest common divisor, which identifies the highest positive integer that divides i and j without any remainder.

3.2.2 Class of Systems

The focus of this research is on nonlinear continuous-time systems featuring multiple inputs and multiple outputs (MIMO), characterized by a collection of nonlinear first-order ordinary differential equations (ODEs) of the form,

$$\dot{x} = F(x, u_{t1}, u_{t2}) = f(x) + g_1(x)u_{t1} + g_2(x)u_{t2} \tag{3.1}$$

The system is described by a state vector $x = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$, a lower-tier control input vector $u_{t1} \in \mathbb{R}^{m_1}$ and an upper-tier control input vector $u_{t2} \in \mathbb{R}^{m_2}$. The system inputs, denoted as u_{t1} and u_{t2}, are bounded by their respective sets $U_1 \subset \mathbb{R}^{m_1}$ and, $U_2 \subset \mathbb{R}^{m_2}$, where $U_1 := \{u_{t1} \in U_1 | u_{t1_{\min,i}} \leq u_{t1_i} \leq u_{t1_{\max,i}}, \forall; i = 1, 2, \cdots, m_1\}$ and $U_2 := \{u_{t2} \in U_2 | u_{t2_{\min,i}} \leq u_{t2_i} \leq u_{t2_{\max,i}}, \forall; i = 1, 2, \cdots, m_2\}$. The quantities $u_{t1_{\min,i}}$ and $u_{t1_{\max,i}}$ correspond to the lowest and highest thresholds for each controlled input in the lower tier, respectively. Similarly, the values $u_{t2_{\min,i}}$ and $u_{t2_{\max,i}}$ pertain to the minimum and maximum values allowed for each controlled input in the upper tier. The functions $f(\cdot)$, $g_1(\cdot)$, and $g_2(\cdot)$ are assumed to be sufficiently smooth vector functions, respectively. For the purpose of simplicity without loss of generality, we introduce the

assumption that $f(0) = 0$, effectively treating the origin as a steady state of Eq. (3.1). For the sake of convenience, we establish the initial time as zero ($t_0 = 0$). Furthermore, the domain of continuous functions that map the interval $[a, b]$ to $\mathbb{R}^n$ is designated as $C([a, b], \mathbb{R}^n)$. Additionally, we define the set $S(\Delta)$ as the assortment of piece-wise constant functions characterized by a period of Δ.

3.2.3 Paillier cryptosystem

In this research, we employ the Paillier cryptosystem [67] to implement encryption and decryption procedures on state measurements of the process (denoted as x) as well as control inputs (represented as u_{t1} and u_{t2}). More importantly, we leverage the semi-homomorphic property of additive homomorphism within the Paillier cryptosystem to conduct linear additive operations within an encrypted space in the lower tier. Similar to numerous other encryption methods, the Paillier cryptosystem's functionality centers on the encryption of plaintext data in the format of natural numbers. The encryption procedure is initiated with the creation of public and private keys. Within the Paillier cryptosystem, integer messages are encrypted to ciphertexts by utilizing the public key during the encryption process. In contrast, the private key facilitates the decryption of ciphertexts, to recover the initial integer messages. The public and private keys are generated as per the following steps:

1. Choose two large prime integers (p and q) randomly, ensuring they meet the requirement $\gcd(pq, (p-1)(q-1)) = 1$.

2. Compute the outcome of multiplying these integers, indicated as $M = pq$.

3. Choose an arbitrary integer g in a manner that $g \in \mathbb{Z}_{M^2}$, with $\mathbb{Z}_{M^2}$ denoting the multiplicative group of integers modulo M^2.

4. Compute $\lambda = \text{lcm}(q-1, p-1)$.

5. Specify $\bar{L}(x) = (x-1)/M$.

6. Verify whether the subsequent modular multiplicative inverse is present:
   $u = (\bar{L}(g^{\lambda} \bmod M^2))^{-1} \bmod M$.

7. Should the inverse not exist, revisit step 3 and opt for an alternate value of g. If the inverse exists, we acquire the public key (M, g) and the private key (λ, u).

Upon acquiring the keys, the public key is disseminated to the intended recipients responsible for carrying out the encryption procedure. Similarly, the private key is shared with the authorized recipients responsible for decrypting the data. Encryption is performed as follows:

$$E_M(m, r) = c = g^m r^M \bmod M^2 \tag{3.2}$$

where r is a randomly selected integer from the set $\mathbb{Z}_M$, and c represents the ciphertext achieved through the encryption of m. The decryption procedure for the ciphertext $c \in \mathbb{Z}_{M^2}$ is executed in the subsequent manner:

$$D_M(c) = m = \bar{L}(c^{\lambda} \bmod M^2) u \bmod M \tag{3.3}$$

3.2.4 Quantization

To utilize the Paillier cryptosystem, it becomes imperative to represent the data to be encrypted as natural numbers, a subset designated as $\mathbb{Z}_M$. However, the signal measurements before encryption are available in the form of floating-point numbers. Consequently, we use the process of quantization to map these floating-point numbers into elements of the set $\mathbb{Z}_M$. To construct this mapping, we use signed fixed-point numbers represented in binary form. The parameters of quantization, namely l_1 and d, signify the total count of bits (integer and fractional) and the number of fractional bits, respectively. Employing these quantization parameters, we create a set denoted as $\mathbb{Q}_{l_1,d}$. This set encompasses rational numbers spanning from -2^{l_1-d-1} to $2^{l_1-d-1} - 2^{-d}$, with each rational number separated by a step of 2^{-d}. A rational number q that resides within the $\mathbb{Q}_{l_1,d}$ set can be articulated as $q \in \mathbb{Q}_{l_1,d}$, where, $\exists \beta \in \{0,1\}^{l_1}$ and $q = -2^{l_1-d-1}\beta_{l_1} + \sum_{i=1}^{l_1-1} 2^{i-d-1}\beta_i$. In order to map a real number data point a onto the $\mathbb{Q}_{l_1,d}$ set, we employ the function $g_{l_1,d}$, illustrated by the equation,

$$\begin{aligned} g_{l_1,d} &: \mathbb{R} \rightarrow \mathbb{Q}_{l_1,d} \\ g_{l_1,d}(a) &:= \arg \min_{q \in \mathbb{Q}_{l_1,d}} |a - q| \end{aligned} \tag{3.4}$$

to acquire the nearest quantized rational number to a specific real number data point. After this, the quantized data is converted into a collection of integers via a one-to-one (bijective) mapping referred to as $f_{l_2,d}$, as described in the work of [19]. This mapping guarantees that the quantized data undergoes a transformation that places it within a subset of the message space $\mathbb{Z}_M$. The

one-to-one mapping can be defined as follows:

$$\begin{aligned} f_{l_2,d} &: \mathbb{Q}_{l_1,d} \to \mathbb{Z}_{2^{l_2}} \\ f_{l_2,d}(q) &:= 2^d q \bmod 2^{l_2} \end{aligned} \tag{3.5}$$

The encryption process involves encrypting integer plaintext messages using the set $Z_{2^{l_2}}$, and the resulting ciphertexts can be decrypted back into the same set $Z_{2^{l_2}}$. Once the upper-tier controller and actuator receive the encrypted signals, the ciphertexts undergo decryption to extract integer plaintext messages that represent quantized states and inputs, respectively. Consequently, it becomes essential to remap these decrypted plaintext messages back to the set $\mathbb{Q}_{l_1,d}$. The inverse mapping, denoted as $f^{-1}_{l_2,d}$, is defined as follows:

$$f^{-1}_{l_2,d} : \mathbb{Z}_{2^{l_2}} \to \mathbb{Q}_{l_1,d} \tag{3.6}$$

$$f^{-1}_{l_2,d}(m) := \frac{1}{2^d} \begin{cases} m - 2^{l_2} & \text{if } m \geq 2^{l_2-1} \\ m & \text{otherwise} \end{cases} \tag{3.7}$$

To illustrate the process of encryption and decryption, we can refer to the example shown in Figure 2.1. For this specific instance, the selected quantization parameters are as follows: $d = 3$, $l_1 = 18$, and $l_2 = 30$. Let us consider the rational number $a = -1.31752$. The impact of quantization is demonstrated in Figure 2.1, where the quantization error, $|a - q| = 0.05748$, is evident.

3.3 Design of the encrypted two-tier control architecture

In the closed-loop framework of the encrypted two-tier control architecture, illustrated in Figure 3.1, the signals, $x_1(t)$ and $x_2(t)$ are transmitted from the sensors to the lower and upper tier, respectively, for the purpose of computing control inputs. The lower and upper tier correspond to the encrypted network tiers 1 and 2 respectively, in Figure 3.1. These signals $x_1(t)$ and $x_2(t)$ undergo encryption using public keys 1 and 2 respectively before they are transmitted to the lower tier consisting of a set of encrypted proportional-integral (PI) controllers and the upper tier comprising a model predictive controller (MPC), respectively. These two tiers operate independently for control input computations, utilizing distinct public and private keys for signal encryption and decryption. Further, both tiers manipulate a distinct set of control inputs, eliminating any concerns related to balancing control signals among actuators. Once the lower tier receives the encrypted data, denoted as c_1, it performs control input calculations within an encrypted space, without decryption, employing additive homomorphic operations. The encrypted control input c_1' is then transmitted to the actuator, where it undergoes decryption using private key 1 to yield the quantized control input $\hat{u}_1(t)$. Concurrently, the upper tier decrypts the ciphertext c_2 and employs the quantized states $\hat{x}_2(t)$ to determine the control input. These quantized states are used to initialize the process model within the MPC at the time t. Following this, the MPC calculates the optimized control inputs $u_2(t)$, which undergo encryption before being transmitted to the actuator. Upon receipt of the encrypted control input c_2', the actuator decrypts it using private key 2, leading to the quantized input $\hat{u}_2(t)$, which is then applied to the process. The presented architecture introduces two potential points within the upper tier where cyberattacks could be initiated: one by manipu-

lating the decrypted state values received by the MPC, and the other by manipulating the control inputs computed by the MPC before encryption. To counteract this vulnerability, an ML-based detector is incorporated at the process site. It intercepts sensor signals prior to their encryption and transmission to the network, thereby ensuring its security. Its role is to detect cyberattacks and subsequently reconfigure the control system in the event of cyberattack detection. This reconfiguration involves deactivating the compromised upper tier and relying solely on the secure, encrypted lower tier to restore the desired closed-loop behavior.

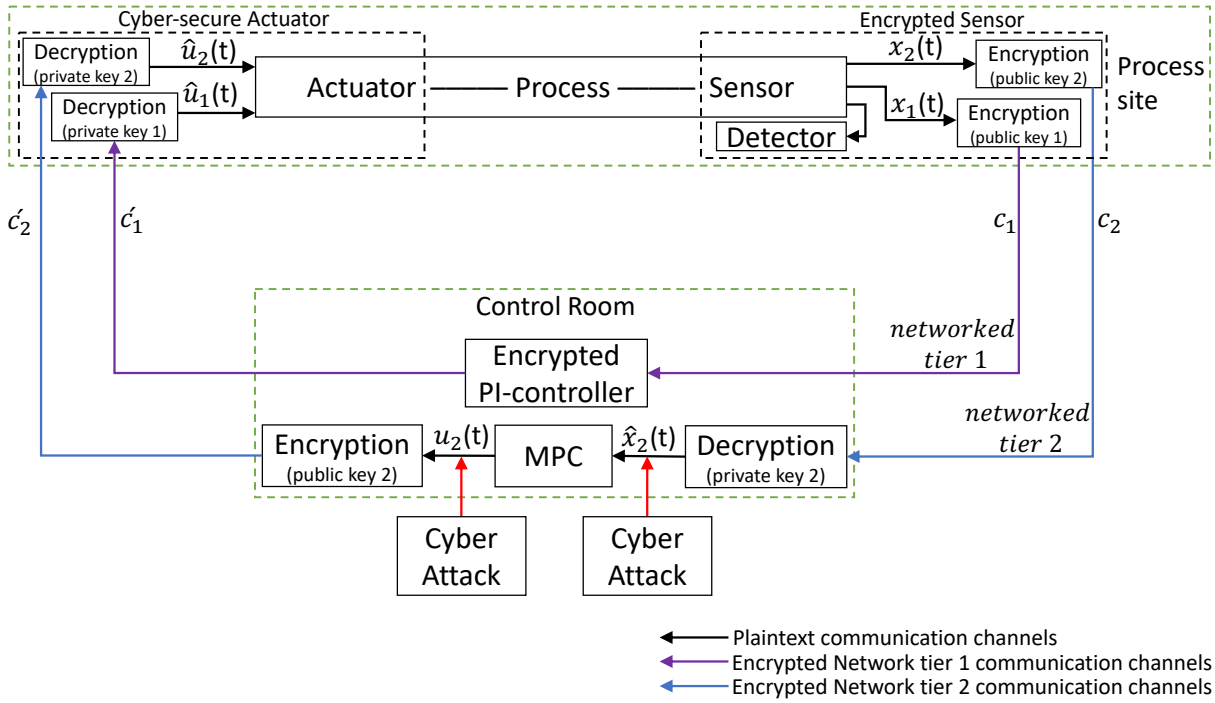

Figure 3.1: Illustration of a two-tier encrypted control scheme.

Remark 3.1. *The encrypted data, in the form of ciphertexts, could potentially be subject to manipulation by an attacker. However, due to the encryption, the attacker gains no information about the process states or the system stability. Any attempts to manipulate the encrypted data would*

lead to significant deviations from actual values. The manipulated encrypted data after decryption could yield infeasible values for certain states, and some control inputs could fall outside the actuation bounds. Such alterations have the potential to destabilize the system, and they can be easily identified by imposing constraints on the control Lyapunov function, eliminating the need for advanced detection techniques. However, in this research, we focus on intelligent cyberattacks that do not force the system out of its stability region. These attacks require the attacker to possess some knowledge about the system and its states, information that can only be obtained through decryption of the states and computation of control inputs before encryption. Therefore, our discussion is centered around these scenarios. Further details regarding the types of cyberattacks launched are provided in Section 3.4. *Additionally, as a proactive measure, a backup control system can be integrated into this design, operating in isolation from any network, to address potential cyberattacks aimed at manipulating encrypted data.*

The presented design of the closed-loop system introduces two types of errors. Initially, there is a quantization error due to the mapping of state data from $\mathbb{R}$ to $\mathbb{Q}_{l_1,d}$ within the sensor-controller communication link. Furthermore, the controller-actuator communication link contributes a control input quantization error as the control input is mapped from a set of real numbers $\mathbb{R}$ to $\mathbb{Q}_{l_1,d}$. Both of these quantization errors are constrained and can be characterized via the mapping equation specified in Eq. (3.4), thereby ensuring that:

$$|x_j(t) - \hat{x_j}(t)| \leq 2^{-d-1} \tag{3.8a}$$

$$|u_k(t) - \hat{u_k}(t)| \leq 2^{-d-1} \tag{3.8b}$$

where d is the quantization parameter used for mapping in Eq. (3.4), while j and k represent the j^{th} state and k^{th} control input, respectively. Taking into account the impact of quantization-induced input errors, the dynamical model under two-tier control architecture employing the nonlinear system of Eq. (3.1) can be expressed as follows:

$$\begin{aligned} \dot{x} = F(x, \hat{u}_{t1}, \hat{u}_{t2}) &= f(x) + g_1(x)\hat{u}_{t1} + g_2(x)\hat{u}_{t2} \\ &= f(x) + g_1(x)(u_{t1} + e_{t1}) + g_2(x)(u_{t2} + e_{t2}) \end{aligned} \tag{3.9}$$

where $e_{t1} = \hat{u}_{t1}(t) - u_{t1}(t)$, $e_{t2} = \hat{u}_{t2}(t) - u_{t2}(t)$ and

$$|e_{ti}| \leq 2^{-d-1} \text{ where } i = \{1, 2\} \tag{3.10}$$

Also, an additional error will be present in the applied control input, as the controller receives $\hat{x}$ instead of the true state x. This error will be confined by the underlying equation, using the local Lipschitz property, where $L_1 > 0$:

$$|\Phi(\hat{x}) - \Phi(x)| \leq L_1|\hat{x} - x| \leq L_1 2^{-d-1} \tag{3.11}$$

Remark 3.2. *Quantization error arises when the desired value to be quantized is not exactly found within the $\mathbb{Q}_{l_1,d}$ set, which comprises quantized values defined by the quantization parameter d. The interval between elements in this set is 2^{-d}. To ascertain the upper limit of this error, let us consider the quantization of a value, x_1. We assume that x_1 falls within the range of y_1 and $y_1 + 2^{-d}$, where y_1 and $y_1 + 2^{-d}$ signify quantized values within $\mathbb{Q}_{l_1,d}$. The quantization procedure involves evaluating the absolute difference between x_1 and y_1 in comparison to that between x_1 and*

$y_1 + 2^{-d}$. When the distance between x_1 and y_1 is less than the distance between x_1 and $y_1 + 2^{-d}$, x_1 is matched with y_1. Alternatively, it is matched with $y_1 + 2^{-d}$. Subsequently, the quantization error is confined within half of the resolution, $\frac{|y_1+2^{-d}-y_1|}{2} = 2^{-d-1}$. This implies that the maximum difference between the quantized value $\hat{x}_1$ and the actual value x_1 is 2^{-d-1}. Thus, selecting a larger quantization parameter, $d \to \infty$, results in a negligible error due to quantization.

3.3.1 Lower-tier encrypted control system

Within the encrypted two-tier control framework, we assume the existence of a feedback controller in the lower tier, represented as $u_{t1} = \Phi(x) \in U_1$, that can attain exponential stability at the origin of the nominal closed-loop system of Eq. (3.1), with $u_{t2} \equiv 0$. This signifies the presence of a $\mathcal{C}^1$ control Lyapunov function $V(x)$ for which the subsequent inequalities are valid across all $x \in \mathbb{R}^n$ within an open region D surrounding the origin:

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2, \tag{3.12a}$$

$$\frac{\partial V(x)}{\partial x} F(x, \Phi(x), 0) \leq -c_3|x|^2, \tag{3.12b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \leq c_4|x| \tag{3.12c}$$

where c_1, c_2, c_3 and c_4 are positive constants. For the nonlinear system described by Eq. (3.1), the region of closed-loop stability can be defined as a level set of the control Lyapunov function V. This stability domain, labeled as Ω_ρ, is defined by $\Omega_\rho := \{x \in D | V(x) \leq \rho\}$, where $\rho > 0$. Hence, originating from any initial condition within Ω_ρ, the applied control input, $\Phi(x)$ guarantees that the system state trajectory, under closed-loop conditions, remains confined within Ω_ρ.

To perform computations in an encrypted space, classical controllers using linear mathematical operations are used to compute control inputs. Specifically, a set of proportional-integral controllers are used. The formula is given as:

$$u(t_k) = K_{c_i}\left(e_i(t_k) + \frac{1}{\tau_i}\int_0^{t_k} e_i(\tau)d\tau\right)\ ,\ e_i(t_k) = y_{sp}(t_k) - y_i(t_k) \tag{3.13}$$

Using the recursive rule to approximate the integral term, the overall controller equation is reformulated using only linear mathematical operations:

$$\begin{aligned} u_{t1_i}(t_k) &= K_{c_i}e_i(t_k) + I_{t_k} \\ &= K_{c_i}e_i(t_k) + K'_{c_i}e_i(t_k) + I_{t_{k-1}} \end{aligned} \tag{3.14}$$

where t_k and t_{k-1} represent the sampling instances k and $k-1$, respectively. u_{t1_i} represents the i^{th} control input in the lower tier, $y_{sp}(t_k)$ and $y_i(t_k)$ represent the set point and state measurement at time t_k, respectively. K_{c_i} and K'_{c_i} represent the gains of the proportional and integral terms, respectively. I_{t_k} represents the integral control action at time t_k. At $k = 0$, I_{t_0} is assumed to be 0.

3.3.2 Upper-tier encrypted model predictive control system

This section formulates the feedback LMPC used in the upper tier of the closed-loop design for the nonlinear system described by Eq. (3.1). Although the LMPC does not compute the control inputs for the lower tier, it estimates their values using the lower tier control law, $u_{t1}(t) = \Phi(\tilde{x}(t))$. This estimation results in a more accurate prediction of the future states of the system, by accounting in the lower-tier control inputs. These predicted state values are used to calculate the LMPC cost function. Accordingly, the upper-tier control inputs that minimize the cost function are com-

puted. Control actions are applied to the nonlinear system using a sample-and-hold approach with a sampling period of Δ [35, 59]. The proposed MPC is formulated in the subsequent manner:

$$\mathcal{J} = \min_{u_{t2} \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), \Phi(\tilde{x}(t)), u_{t2}(t))dt \tag{3.15a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F(\tilde{x}(t), \Phi(\tilde{x}(t)), u_{t2}(t)) \tag{3.15b}$$

$$u_{t2}(t) \in U_2, \ \forall \, t \in [t_k, t_{k+N}) \tag{3.15c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{3.15d}$$

$$\dot{V}(\hat{x}(t_k), \Phi(\hat{x}(t_k)), u_{t2}(t_k)) \leq \dot{V}(\hat{x}(t_k), \Phi(\hat{x}(t_k)), 0), \text{ if } \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{\min}} \tag{3.15e}$$

$$V(\tilde{x}(t)) \leq \rho_{\min}, \ \forall \, t \in [t_k, t_{k+N}), \text{ if } \hat{x}(t_k) \in \Omega_{\rho_{\min}} \tag{3.15f}$$

The predicted state trajectory of the LMPC process model is represented as $\tilde{x}$. The quantized states, $\hat{x}$, serve as the initial conditions for the LMPC process model to predict the state trajectories. The number of sampling periods within the prediction horizon is represented as N. The LMPC algorithm computes the optimal input sequence $u_{t2}^*(t|t_k)$ for the entire prediction horizon $t \in [t_k, t_{k+N})$ but transmits only the first input of this sequence to the actuator for application to the system within the interval $t \in [t_k, t_{k+1})$. The rationale behind predicting state trajectories for extended durations compared to the control input application period by the actuator is to optimize the existing control inputs. This optimization aims to minimize the control cost function not only within the current sampling period but also over the prediction horizon, thereby enhancing overall performance.

The encrypted LMPC method employs a sequence of specific actions: it uses quantized states $\hat{x}$ to predict the trajectory of the system states as per Eq. (3.15b), which is used to integrate the

cost function of Eq. (3.15a) to calculate optimized control inputs for the entire prediction horizon. The actuator applies only the control inputs of the first sampling period, and this process is iterated at each sampling period. Eq. (3.15c) represents the constraints imposed on the control inputs. The constraint in Eq. (3.15d) uses the quantized states (after decryption) to initialize the plant model described in Eq. (3.15b). If the state $x(t_k)$ at time t_k lies within the set $\Omega_{\hat{\rho}} \setminus \Omega_{\rho_{\min}}$, where $\rho_{\min}$ represents a level set of V in proximity to the origin, the Lyapunov constraint outlined in Eq. (3.15e) ensures that the time-derivative of the control Lyapunov function of the closed-loop system under the two-tier control scheme is less than or equal to the time-derivative of the control Lyapunov function when the system is controlled by only the lower tier. When the closed-loop state $x(t_k)$ enters $\Omega_{\rho_{\min}}$, the constraint detailed in Eq. (3.15f) ensures that this state remains within $\Omega_{\rho_{\min}}$.

3.3.3 Lower-tier stability under encryption

Given the occurrence of quantization errors in the links between sensors and controllers, as well as controllers and actuators, it becomes imperative to delineate a region of closed-loop stability, denoted as $\Omega_{\hat{\rho}}$, which is encompassed within the broader Ω_ρ (specifically, $\hat{\rho} < \rho$). The subsequent theorem establishes that the encrypted lower-tier controller $\Phi(\hat{x}) \in U_1$ can achieve exponential stability at the origin for the nonlinear system introduced in Eq. (3.9).

Theorem 3.1. *Let us consider the nonlinear system introduced in Eq.* (3.9)*, which can be represented as* $\dot{x} = F(x, \hat{u}_{t1}, 0)$ *when exclusively under lower-tier encrypted control. The initial state is* $x_0 \in \Omega_{\hat{\rho}}$*, and the stabilizing control law is denoted as* $u_{t1} = \Phi(x) \in U_1$*. Consequently, the equilibrium point of the closed-loop system derived from Eq.* (3.9) *through encrypted control becomes practically stable for all* $x_0 \in \Omega_{\hat{\rho}}$*. In this context, the closed-loop state* $x(t)$ *remains within* Ω_ρ *for*

all instances, and the ensuing inequalities remain valid:

$$\dot{V} \leq -c_5|x|^2 \quad \forall |x| \geq \frac{c_4(L_1+1)2^{-d-1}}{c_3\theta} = \mu \tag{3.16a}$$

$$\limsup_{t\to\infty} |x| \leq b \tag{3.16b}$$

where d is the quantization parameter, $c_3, c_4, L_1 > 0$, b is a positive constant (which can be expressed as a class $\mathcal{K}$ function of μ), $0 < \theta < 1$ and $c_5 = (1-\theta)c_3$.

Proof. Based on the nonlinear system of Eq. (3.9), the time-derivative of V can be written as:

$$\begin{aligned}
\dot{V} &= \frac{\partial V}{\partial x} F(x, \hat{u}_{t1}, 0) \\
&= \frac{\partial V}{\partial x} F(x, u_{t1} + e_1, 0) \\
&= \frac{\partial V}{\partial x} F(x, \Phi(\hat{x}) + e_1, 0) \\
&= \frac{\partial V}{\partial x} F(x, \Phi(\hat{x}) + e_1, 0) - \frac{\partial V}{\partial x} F(x, \Phi(x), 0) + \frac{\partial V}{\partial x} F(x, \Phi(x), 0)
\end{aligned} \tag{3.17}$$

Based on Eq. (3.12b), it follows that

$$\begin{aligned}
\dot{V} &\leq \frac{\partial V}{\partial x} F(x, \Phi(\hat{x}) + e_1, 0) - \frac{\partial V}{\partial x} F(x, \Phi(x), 0) - c_3|x|^2 \\
&= \frac{\partial V}{\partial x} \big(f(x) + g_1(x)(\Phi(\hat{x}) + e_1)\big) - \frac{\partial V}{\partial x} \big(f(x) + g_1(x)(\Phi(x))\big) - c_3|x|^2 \\
&= \frac{\partial V}{\partial x} \big(f(x) + g_1(x)(\Phi(\hat{x}) + e_1) - f(x) - g_1(x)(\Phi(x)\big) - c_3|x|^2 \\
&= \frac{\partial V}{\partial x} \big(g_1(x)(\Phi(\hat{x}) - \Phi(x))\big) + \frac{\partial V}{\partial x} g_1(x) e_1 - c_3|x|^2
\end{aligned} \tag{3.18}$$

Applying the inequalities of Eq. (3.12c), Eq. (3.10) and Eq. (3.11), it follows that

$$\begin{aligned}\dot{V} &\leq c_4|x|L_1 2^{-d-1} + c_4|x|2^{-d-1} - c_3|x|^2 \\ &= -c_3|x|^2 + c_4|x|(L_1+1)2^{-d-1} \\ &= -(1-\theta)c_3|x|^2 - \theta c_3|x|^2 + c_4|x|(L_1+1)2^{-d-1}\end{aligned} \tag{3.19}$$

Therefore, if the condition of Eq. (3.16a) on $|x|$ is satisfied i.e., $|x| \geq \frac{c_4(L_1+1)2^{-d-1}}{c_3\theta} = \mu$, it follows that

$$\begin{aligned}\dot{V} &\leq -(1-\theta)c_3|x|^2 \leq 0 \\ &\leq -c_5|x|^2 \leq 0\end{aligned} \tag{3.20}$$

where $c_5 = (1-\theta)c_3$. Thus, based on Eq. (3.20), we have that $\dot{V}$ is negative for all $x \in \Omega_{\hat{\rho}}$ that satisfy the condition of Eq. (3.16a).

Given that $\Omega_{\hat{\rho}}$ is a level set of V and its derivative, $\dot{V}$, is negative for all $x \in \Omega_{\hat{\rho}}$, it can be inferred that the state of the closed-loop system, denoted as $x(t)$, remains within $\Omega_{\hat{\rho}}$ throughout all time. Moreover, referencing Theorem 4.18 in [44], it can be deduced that:

$$\limsup_{t\to\infty} |x(t)| \leq b \tag{3.21}$$

Hence, as the quantization parameter $d \to \infty$, following the definition of μ from Eq. (3.16a), $\mu \to 0$ and, therefore, the ultimate bound approaches zero, proving that larger values of the quantization parameter d results in a smaller error between the state and input trajectories of the encrypted control system and the non-encrypted control system. This proves that the closed-loop states of the nonlinear system of Eq. (3.9) are ultimately bounded under the stabilizing controller $u_{t1} =$

$\Phi(\hat{x}) \in U_1$ for sufficiently large d. □

3.3.4 Two-tier stability under encryption

Theorem 3.2. *Taking into consideration the two-tier encrypted control architecture for the system of Eq.* (3.9)*, we examine its behavior within the context of the closed-loop encrypted LMPC design detailed in Eq.* (3.15) *for the upper tier. This design relies on a stabilizing lower-tier controller denoted as* $u_{t1} = \Phi(\hat{x}) \in U_1$*, which adheres to the inequalities outlined in Eq.* (3.12)*. Furthermore, we assume that the initial state* x_0 *resides within the region* $\Omega_{\hat{\rho}}$*. For the purpose of our analysis, we introduce* $\Delta > 0$*,* $\epsilon_w > 0$*, and parameters* $\hat{\rho} > \rho_{\min} > \rho_s$ *that fulfill the following conditions.*

$$\begin{aligned} &-\frac{c_3}{c_2}\rho_s + L_x^{'} M_F \Delta + L_u^{'} \delta \leq -\epsilon_w \\ &\rho_{\min} = \max\{V(x(t+\Delta)) | V(x(t)) \leq \rho_s\} \end{aligned} \tag{3.22}$$

Then, the closed-loop state $x(t)$ *remains bounded in* $\Omega_{\hat{\rho}}$ *and is ultimately bounded in* $\Omega_{\rho_{\min}}$.

Proof. Consider the state $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_s}$. The time-derivative of V under the control inputs calculated by the LMPC of Eq. (3.15) for the nonlinear system of Eq. (3.9) at t_k can be written as:

$$\begin{aligned} \dot{V} &= \frac{\partial V(x(t))}{\partial x} F\big(x(t), \Phi(x(t_k)) + e_{t1}, u_{t2}(t_k) + e_{t2}\big) \\ \dot{V} &= \frac{\partial V(x(t_k))}{\partial x} F\big(x(t_k), \Phi(x(t_k)), u_{t2}(t_k)\big) \\ &\quad + \frac{\partial V(x(t))}{\partial x} F\big(x(t), \Phi(x(t_k)) + e_{t1}, u_{t2}(t_k) + e_{t2}\big) \\ &\quad - \frac{\partial V(x(t_k))}{\partial x} F\big(x(t_k), \Phi(x(t_k)), u_{t2}(t_k)\big) \end{aligned} \tag{3.23}$$

for all $t \in [t_k, t_{k+1}]$. Here, e_{t1} and e_{t2} represent the error in the control inputs of the lower and upper tiers, respectively, due to quantization. Based on Eqs. (3.18) and (3.19), the error e_{t1} can

be bounded by $(L_1+1)2^{-d-1}$. Similarly, the error e_{t2} can be bounded by $\eta 2^{-d-1}$. Based on the inequality of Eq. (3.12b), it follows from Eq. (3.23) that:

$$\begin{aligned}\dot{V} \leq & -c_3|x(t_k)|^2 + \frac{\partial V(x(t))}{\partial x} F\big(x(t), \Phi(x(t_k)) + e_{t1}, u_{t2}(t_k) + e_{t2}\big) \\ & - \frac{\partial V(x(t_k))}{\partial x} F\big(x(t_k), \Phi(x(t_k)), u_{t2}(t_k)\big)\end{aligned} \tag{3.24}$$

In the encrypted LMPC, the constraint of Eq. (3.15e) ensures that, if $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_{\min}}$, then the closed-loop state is driven towards the origin at t_{k+1} (to a lower level set of V). Based on the fact that the errors $|e_{t1}|$ and $|e_{t2}|$ are bounded, using the Lipschitz condition and the inequality of Eq. (3.12a), it follows from Eq. (3.24) that:

$$\dot{V} \leq -\frac{c_3}{c_2}\rho_s + L_x^{'}|x(t) - x(t_k)| + L_{u1}^{'}(L_1+1)2^{-d-1} + L_{u2}^{'}\eta 2^{-d-1} \tag{3.25}$$

where $L_x^{'}, L_{u1}^{'}, L_{u2}^{'} > 0$. Due to the continuity of $x(t)\ \forall\ t \in [t_k, t_{k+1})$, we can write that $|x(t) - x(t_k)| \leq M_F\Delta\ \forall\ t \in [t_k, t_{k+1})$. Using this bound, it follows from Eq. (3.25) that:

$$\dot{V} \leq -\frac{c_3}{c_2}\rho_s + L_x^{'}M_F\Delta + L_u^{'}\delta \tag{3.26}$$

where $L_u^{'} = \big(L_{u1}^{'}(L_1+1) + L_{u2}^{'}\eta\big)$ is a positive constant. $\delta = 2^{-d-1}$ is also a positive constant, dependent on the quantization parameter d selected. As evident, the magnitude of the error due to quantization, represented by the last term of Eq. (3.26), will be smaller as $d \to \infty$. Hence, selecting a higher quantization parameter is advisable whenever possible. Thus, if $-\frac{c_3}{c_2}\rho_s + L_x^{'}M_F\Delta + L_u^{'}\delta \leq -\epsilon_w$, then $\dot{V} \leq -\epsilon_w$ for any $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_s}$. This establishes that, if the conditions of Eq. (3.22) are met, the state of the closed-loop system is always bounded in $\Omega_{\hat{\rho}}$, and it ultimately converges

to $\Omega_{\rho_s} \subseteq \Omega_{\rho_{\min}}$ and remains there. □

3.4 Cyberattack types and machine learning-based detection

The upper-tier control system, where encrypted sensor signals are decrypted upon receipt and further transmitted to the MPC in decrypted form, could be susceptible to cyberattacks. Similarly, the control inputs computed by the MPC prior to encryption might also face vulnerability to cyber threats. These signals, transmitted in plaintext, could potentially be manipulated by an attacker if the control room responsible for decryption and encryption lacks full cyber-physical security.

In contrast, the lower tier receives encrypted signals and calculates control inputs within an encrypted space, transmitting them without decryption within the lower networked-tier. This approach ensures complete security, even if the control room receiving and transmitting the encrypted signals is not entirely secure, as the data remains encrypted throughout the networked communication within the lower tier. Also, as depicted in Figure 3.1, where the encrypted network tiers 1 and 2 correspond to the lower and upper tiers, respectively, the lower tier does not necessitate sharing access to its public and private keys with the control room, in contrast to the upper tier. This distinction contributes to the enhanced cybersecurity of the lower tier, even in situations where the security of the control room might be compromised. Upon detecting a cyberattack, the upper-tier control system is disabled, while only the lower-tier control system remains operational. The latter is capable of stabilizing the system at its steady state.

3.4.1 Types of Cyberattacks

Given the adaptability of intelligent cyberattacks to process and control system behaviors, it is assumed that these attacks possess the potency to access information regarding the stability region of the two-tier controlled process. The scope of cyberattacks in the encrypted two-tier control architecture typically encompasses manipulation of signal data, where data received by the MPC and the control inputs computed by it could potentially be subjected to tampering. This study addresses attacks directed at both the sensor signals received by the MPC and the control inputs computed by it.

In regular operational scenarios, decrypted sensor signals accurately reflect the true state data. However, if this data is tampered with, it can lead to control actions driving the process away from its steady state. Likewise, manipulation of control inputs can deviate process states by withholding the necessary control action. Intelligent cyberattacks are designed in a manner such that, when launched on sensor signals, the controller is capable of calculating an appropriate control action, within the actuation bounds, using the attacked state. Similarly, when launched on control inputs, the manipulated data avoids falling beyond actuation limits, thereby evading detection by conventional mechanisms. To address these challenges, advanced machine learning algorithms utilizing neural networks are used for cyberattack detection. Some commonly launched attacks are considered below.

Min-Max cyberattack

Min-Max cyberattacks are specifically crafted to maximize destabilizing impact within the shortest timeframe while evading detection. To maintain their concealment from conventional detection

methods, min-max attacks target the lower value of the following two conditions:

1. A window around equilibrium: This condition centers around a window encompassing the equilibrium point of the affected state(s), representing a range of realistic physical operational conditions.

2. Extreme state values: The second condition revolves around state values situated farthest from the equilibrium point, whether they are minimum or maximum values. The intention is to ensure that the system remains within the closed-loop stability region Ω_ρ.

By introducing attacks based on the aforementioned conditions, it is guaranteed that the state measurements received by the controller after manipulation remain inside the stability region delineated by the configured operational window. Furthermore, these attacks circumvent setting off any conventional detection alarms rooted in boundary values.

The formulation of the min-max attack is expressed in the following manner:

$$\bar{x}(t_i) = \min \left\{ \arg\max_{x \in \mathbb{R}^n} \{V(x(t_i))_{\leq} \rho\}, \arg\max_{x \in \mathbb{R}^n} \{x(t_i) \in \chi\} \right\}, \forall i \in [i_o, i_o + L_a] \tag{3.27a}$$

$$\bar{u}_{t2}(t_i) = \min \left\{ \arg\max_{u_{t2} \in \mathbb{R}^{m_2}} \{V(x(t_i))_{\leq} \rho\}, \arg\max_{u_{t2} \in \mathbb{R}^{m_2}} \{u_{t2}(t_i) \in U_2\} \right\}, \forall i \in [i_o, i_o + L_a] \tag{3.27b}$$

where ρ defines the region of the Lyapunov function $V(x)$ that characterizes the stability boundaries of the closed-loop system under the two-tier control architecture. The notation $\chi = \{x_l \leq x \leq x_u\}$ represents the desired operating range for the system states, where x represents the compromised sensor signals to be received by the MPC after decryption at each time step. The value i_o marks the time when the attack is introduced, and L_a denotes the duration of the attack in terms of

sampling periods. Similarly, the symbol $u_{t2}(t_i)$ signifies the control input that has been tampered with before encryption. The symbols $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ correspond to the altered or manipulated values of the sensor signal and control input of the upper tier, respectively.

Replay cyberattack

In a replay attack, the attacker initially captures portions of the system output aligned with a regular operational state marked by substantial oscillations. Subsequently, the attacker intervenes to intercept and restore the present process state measurements to the previously recorded values. Replay attacks can be represented using the subsequent equations:

$$\bar{x}(t_i) = x(t_k),\ \forall k \in [k_o, k_o + L_a],\ \forall i \in [i_o, i_o + L_a] \tag{3.28a}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_k),\ \forall k \in [k_o, k_o + L_a],\ \forall i \in [i_o, i_o + L_a] \tag{3.28b}$$

where $x(t_k)$ and $u_{t2}(t_k)$ are the true plant measurement and control input, respectively. L_a denotes the extent of the attack as measured in terms of sampling intervals. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ denote the sequence of replay attacks initiated at time t_{i_o} by duplicating prior plant measurements and control inputs recorded commencing from time t_{k_o}. As the previous plant outputs are derived from authentic closed-loop measurements and obtained via secure sensors, these state values are hypothesized to fall within the stability region and operating bounds. Consequently, by reproducing these values and reintroducing them into the controller, conventional detectors are unlikely to detect the anomaly.

False-data-injection cyberattack

False-data-injection (FDI) cyberattacks involve the insertion of fabricated information into authentic data. This intrusion does not necessitate familiarity with previous event data or system specifics. Introducing deceptive data such that $V(x) \leq \rho$ might not lead to system destabilization, but could merely modify its operational state based on the process dynamics, rendering them challenging to identify through conventional alarm threshold approaches. FDI attacks are represented as follows:

$$\bar{x}(t_i) = x(t_i) + \nu, \ \forall i \in [i_o, i_o + L_a] \tag{3.29a}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_i) + \nu, \ \forall i \in [i_o, i_o + L_a] \tag{3.29b}$$

where $x(t_i)$ and $u_{t2}(t_i)$ are the true plant measurement and control input, respectively. ν represents the false data injected. L_a represents the length of the attack in terms of sampling periods. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ are the FDI attacks introduced from time t_{i_o} up to time $t_{i_o+L_a}$.

Sinusoidal cyberattack

Sinusoidal attack constitutes a form of cyberattack involving the introduction of a sinusoidal signal into authentic data. Due to the inherent periodic oscillations in a sinusoidal function, these attacks can be challenging to identify, as they lack the potential to destabilize the system while inducing substantial fluctuations. Moreover, their periodic pattern can evade standard detection mechanisms.

Their representation can be expressed as follows:

$$\bar{x}(t_i) = x(t_i) + a\sin(2\pi k t_i),\ \forall i \in [i_o, i_o + L_a] \tag{3.30a}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_i) + a\sin(2\pi k t_i),\ \forall i \in [i_o, i_o + L_a] \tag{3.30b}$$

where $x(t_i)$ and $u_{t2}(t_i)$ are the true plant measurement and control input, respectively. k and a are constants. L_a represents the length of the attack in terms of sampling periods. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ are the sinusoidal attacks introduced from time t_{i_o} up to time $t_{i_o+L_a}$.

Surge cyberattack

Surge cyberattack is a stealthy cyberattack that cannot be detected by conventional methods such as cumulative sum (CUMSUM). Surge attacks share similarities with min-max attacks in their initial behavior of maximizing disruptive impact over a brief interval before diminishing to a lower level. In our scenario, the initial duration of the surge, measured in sampling periods, is denoted as L_s and is chosen to be between 2 and 5 inclusive. This choice helps distinguish surge attacks from min-max attacks, as the surge exhibits distinct characteristics during its latter phase. After the sampling duration, L_s, a bounded noise is introduced to the genuine data, resembling the approach

used in a false-data-injection attack. Their representation can be expressed as follows:

$$\bar{x}(t_i) = \min\left\{\arg\max_{x\in\mathbb{R}^n}\{V(x(t_i))_{\leq}\rho\}, \arg\max_{x\in\mathbb{R}^n}\{x(t_i)\in\chi\}\right\}, \forall i \in [i_o, i_o + L_s] \tag{3.31a}$$

$$\bar{x}(t_i) = x(t_i) + \eta(t_i),\ \forall i \in (L_s, i_o + L_a] \tag{3.31b}$$

$$\bar{u}_{t2}(t_i) = \min\left\{\arg\max_{u_{t2}\in\mathbb{R}^{m_2}}\{V(x(t_i))_{\leq}\rho\}, \arg\max_{u_{t2}\in\mathbb{R}^{m_2}}\{u_{t2}(t_i)\in U_2\}\right\},$$
$$\forall i \in [i_o, i_o + L_s] \tag{3.31c}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_i) + \eta(t_i),\ \forall i \in (L_s, i_o + L_a] \tag{3.31d}$$

where $x(t_i)$ and $u_{t2}(t_i)$ are the true plant measurement and control input of the upper tier, respectively. The initial surge corresponds to Eqs. (3.31a) and (3.31c), while the subsequent noise addition is represented by Eqs. (3.31b) and (3.31d). $\eta_l \leq \eta(t_i) \leq \eta_u$ is the bounded noise added to the data following the initial surge. L_a represents the length of the attack in terms of sampling periods. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ are the surge attacks introduced from time t_{i_o} up to time $t_{i_o+L_a}$.

Geometric cyberattack

Geometric cyberattacks adhere to a strategy that gradually erodes the stability of the closed-loop system. It initiates with a gradual decay, which then accelerates exponentially as time progresses. This attack type attains its highest impact as the attack duration concludes. The initial move of the attacker involves introducing a constant value, labeled as β, to the genuine data (ensuring β remains considerably lower than the threshold value set within a min-max attack). In each subsequent time step, this initial deviation is magnified by a factor of $(1 + \alpha)$, where α falls within the range $(0, 1)$, until it reaches the maximum allowable attack magnitude. The two parameters, α and β,

are prudently selected while accounting for the stability region, operational boundaries, and attack duration. Geometric attacks can be formulated as follows:

$$\bar{x}(t_i) = x(t_i) + \beta \times (1+\alpha)^{i-i_o},\ \forall i \in [i_o, i_o + L_a] \tag{3.32a}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_i) + \beta \times (1+\alpha)^{i-i_o},\ \forall i \in [i_o, i_o + L_a] \tag{3.32b}$$

where the parameters α and β define the speed and magnitude of the geometric attack. $x(t_i)$ and $u_{t2}(t_i)$ are the true plant measurement and control input of the upper tier, respectively. L_a represents the length of the attack in terms of sampling periods. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ are the geometric attacks introduced from time t_{i_o} up to time $t_{i_o+L_a}$.

3.4.2 Machine-Learning-based cyberattack detection

Utilizing a data-driven approach to construct the cyberattack detector offers numerous advantages. Firstly, given the potential access of attackers to process-behavior information, traditional first-principles model-based detection methods relying on predetermined statistical thresholds and false alarm biases become inadequate. Secondly, in real-world scenarios, the structure and parameters of the plant model are susceptible to alterations due to evolving operational conditions. In this context, adopting a data-centric approach for training the cyberattack detection mechanism proves resilient against both dynamic process changes and intricately crafted attacks.

In the realm of well-established machine learning approaches, neural networks (NN) have showcased their effectiveness in both supervised and unsupervised classification scenarios. In this particular study, we focus on a supervised classification task employing a two-class classification

framework to determine whether a cyberattack has impacted the upper-tier control system.

When attacks involve data manipulation, they can manifest in various forms or patterns. Building a model to classify attack types can lead to increased computational demands and model intricacy. Since our primary aim is to ascertain whether the upper-tier control has been subjected to an attack or not, we opt for a binary classification model. This approach simplifies the task and facilitates the identification of attack occurrences. Furthermore, to evaluate the effectiveness of the detector against attack patterns it has not encountered during training and validation, we introduce additional attack scenarios in the testing set that differ from those it has been exposed to previously.

The adopted neural network involves a sequence of nonlinear transformations, where neurons in the first hidden layer are computed from input data. Subsequent hidden neurons are derived from their preceding layer, culminating in the output being computed from neurons in the final hidden layer. These transformations occur in the form of activation functions involving biases and the weighted sum of inputs (or neurons from the previous layer). The fundamental structure of the utilized neural network model is depicted in Figure 3.2, where each input corresponds to the feature-wise normalized control Lyapunov function computed from state measurements across a sequence of sampling instances. The control Lyapunov function captures the dynamics of all states of the system, making it an effective one-dimensional input feature for attack detection. While training the model, to make it generic, and to prevent overfitting, we adopted the standard practice of normalizing the training, testing, and validation datasets. Hence, while supplying the control Lyapunov function data during operation, this is normalized with respect to the mean and standard deviation of the training dataset, which is calculated prior to implementation of the detector in the process. This approach aids in aligning the data distributions and mitigates the influence of varying

scales across features, thereby facilitating model training and enhancing model performance. The resulting output vector denotes the predicted class label, distinguishing between "cyberattack" and "no attack". The mathematical representation of the feed-forward neural network with two hidden layers can be formulated as:

$$\theta_j^{(1)} = g_1\left(\sum_{i=1}^{N_T} w_{ij}^{(1)}\hat{V}(x(t_i)) + b_j^{(1)}\right) \tag{3.33a}$$

$$\theta_j^{(2)} = g_2\left(\sum_{i=1}^{h_1} w_{ij}^{(2)}\theta_i^{(1)} + b_j^{(2)}\right) \tag{3.33b}$$

$$\theta_j^{(3)} = g_3\left(\sum_{i=1}^{h_2} w_{ij}^{(3)}\theta_i^{(2)} + b_j^{(3)}\right) \tag{3.33c}$$

$$\mathrm{y}_{\mathrm{pred}} = \left[\theta_1^{(3)}, \theta_2^{(3)}, ..., \theta_H^{(3)}\right]^\top \tag{3.33d}$$

where $\theta_j^{(1)}$, $\theta_j^{(2)}$, and $\theta_j^{(3)}$ denote the output of the j^{th} neuron of the first hidden layer, the second

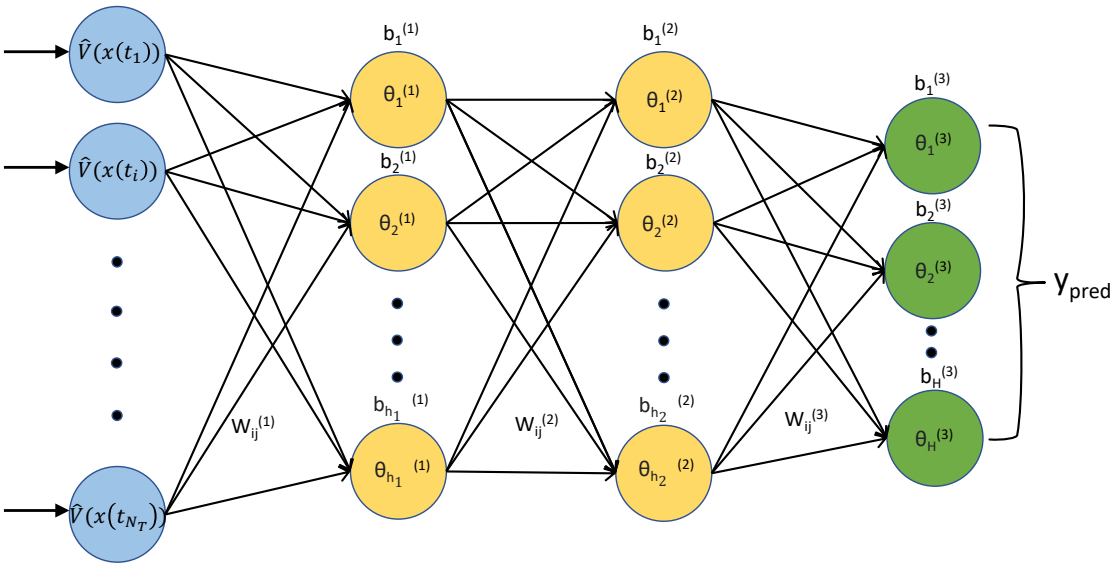

Figure 3.2: Feed-forward neural network structure of the proposed ML-based cyberattack detector.

hidden layer, and the output layer, respectively. h_1 and h_2 stand for the neuron counts in the first and second hidden layers, while H signifies the number of class labels, equal to the number of

neurons in the output layer. Within the input layer, the normalized control Lyapunov function of the complete state measurements at time t_i, denoted as $\hat{V}(x(t_i))$, serves as the input variable. The index $i = 1, .., N_T$, with N_T being the duration of the time-varying trajectory for each input sample. The connections between neurons i and j in successive layers are weighted by $w_{ij}^{(k)}$, where $k = 1, 2, 3$. Additionally, the bias applied to the j^{th} neuron in the k^{th} layer is represented as $b_j^{(k)}$. Each layer receives input from its preceding layer and processes the input with optimized weights, biases, and nonlinear activation functions, represented by g_k. Within the output layer, the vector $\mathrm{y}_{\mathrm{pred}}$ provides the probabilities for each class label concerning the analyzed sample. Notably, the neuron with the highest probability signifies the predicted class label.

The process of calculating training and testing accuracies entails computing the proportion of accurately classified samples relative to the total number of samples present within their respective training and testing datasets. In the development of a neural network model for cyberattack detection, closed-loop values of the control Lyapunov function are gathered over a fixed duration (N_T samples), encompassing various randomly initialized initial conditions. This is done both within and beyond the stability region Ω_ρ, ensuring coverage of a wide spectrum of allowable conditions. Given that $V(x)$ captures the dynamic characteristics of all states, it serves as an effective one-dimensional input feature for the attack detection problem. To improve training accuracy, an equivalent number of samples from each class are assembled. Each sample corresponds to a distinct set of initial conditions for the closed-loop system simulation. Further details of the model such as number of input neurons, activation functions, training, validation and testing accuracies are reported in Section 3.5.3.

Remark 3.3. *To distinguish between dynamics in the control Lyapunov function caused by process fluctuations and cyberattacks, Gaussian-distributed noise is introduced into sensor signal measurements of the training and testing datasets. This accounts for both sensor noise and process disturbances, aiding the model to discern cyberattacks from fluctuations. In addition, a sliding window alarm is implemented, whereby the upper tier is deactivated only if the model identifies a cyberattack in three out of four consecutive sampling instances. This mechanism prevents accidental deactivation of the upper tier due to inherent process disturbances and ensures that, if inadvertently a cyberattack is detected at a single sampling instance due to process disturbances, the upper tier remains active. Such strategies are pivotal for the accurate differentiation of cyberattacks from process fluctuations.*

3.5 Application to a chemical process

This section showcases the practical application of the suggested encrypted two-tier control framework in the context of a large-scale chemical process. We develop a nonlinear dynamical model based on first-principles modeling fundamentals. Subsequently, we employ it as the basis for constructing a first-principles-based encrypted LMPC. Alongside this, a set of encrypted PI controllers, capable of computing control input in an encrypted space, is formulated, and the control architecture is augmented with an ML-based cyberattack detector. Subsequently, we perform closed-loop simulations using the first-principles-based process model. Throughout these simulations, various cyberattacks are initiated, leading to the examination of multiple detection and control scenarios.

3.5.1 Process description and model development

The process considered is the synthesis of ethylbenzene (EB) through the conversion of ethylene (E) and benzene (B). The primary reaction, termed as "primary", is characterized as a second-order, exothermic, and irreversible reaction, in conjunction with two supplementary side reactions. These reactions occur within two non-isothermal, well-mixed continuous stirred tank reactors (CSTRs). The chemical reactions taking place are articulated as follows:

$$\mathrm{C_2H_4} + \mathrm{C_6H_6} \rightarrow \mathrm{C_8H_{10}} \qquad \text{(primary)} \tag{3.34a}$$

$$\mathrm{C_2H_4} + \mathrm{C_8H_{10}} \rightarrow \mathrm{C_{10}H_{14}} \tag{3.34b}$$

$$\mathrm{C_6H_6} + \mathrm{C_{10}H_{14}} \rightarrow 2\mathrm{C_8H_{10}} \tag{3.34c}$$

The state variables are the concentration of ethylene, benzene, ethylbenzene, di-ethylbenzene, and the reactor temperature for each $\mathrm{CSTR}_i,\ i = (1, 2)$, in deviation terms, that is: $x^\top = [C_{E_1} - C_{E_{1s}},\ C_{B_1} - C_{B_{1s}},\ C_{EB_1} - C_{EB_{1s}},\ C_{DEB_1} - C_{DEB_{1s}},\ T_1 - T_{1s},\ C_{E_2} - C_{E_{2s}},\ C_{B_2} - C_{B_{2s}},\ C_{EB_2} - C_{EB_{2s}},\ C_{DEB_2} - C_{DEB_{2s}},\ T_2 - T_{2s}]$. The subscript "$s$" denotes the steady-state value. The rate of heat removal for the two reactors $[Q_1 - Q_{1s},\ Q_2 - Q_{2s}]$ are the control inputs manipulated by the lower tier using encrypted PI controllers, which are bounded by the closed sets, $[-10^4\ \mathrm{kW},\ 2 \times 10^3\ \mathrm{kW}]$ and $[-1.5 \times 10^4\ \mathrm{kW},\ 5 \times 10^3\ \mathrm{kW}]$ respectively. The inlet feed concentrations for each reactor, $[C_{E_{o1}} - C_{E_{o1s}},\ C_{B_{o1}} - C_{B_{o1s}},\ C_{E_{o2}} - C_{E_{o2s}},\ C_{B_{o2}} - C_{B_{o2s}}]$, are the control inputs manipulated by the upper tier using an encrypted MPC, which are bounded by the closed sets $[-2.5\,\mathrm{kmol/m^3},\ 2.5\,\mathrm{kmol/m^3}]$, $[-2.5\,\mathrm{kmol/m^3},\ 2.5\,\mathrm{kmol/m^3}]$, $[-3\,\mathrm{kmol/m^3},\ 3\,\mathrm{kmol/m^3}]$, and $[-3\,\mathrm{kmol/m^3},\ 3\,\mathrm{kmol/m^3}]$, respectively. The control objective is to maintain the operation

of both CSTRs at their unstable equilibrium state through the utilization of the encrypted two-tier control scheme, employing quantized states and inputs for computation and actuation. Through the application of mass and energy balance principles, the foundational model for the CSTRs is constructed. An illustrative visualization of this model is presented in Figure 3.3. In particular, the dynamic representation of the initial CSTR is captured by the subsequent set of ordinary differential equations (ODEs):

$$\frac{\mathrm{d}C_{E_1}}{\mathrm{d}t} = \frac{F_1 C_{Eo_1} - F_{out_1} C_{E_1}}{V_1} - r_{1,1} - r_{1,2} \tag{3.35a}$$

$$\frac{\mathrm{d}C_{B_1}}{\mathrm{d}t} = \frac{F_1 C_{Bo_1} - F_{out_1} C_{B_1}}{V_1} - r_{1,1} - r_{1,3} \tag{3.35b}$$

$$\frac{\mathrm{d}C_{EB_1}}{\mathrm{d}t} = \frac{-F_{out_1} C_{EB_1}}{V_1} + r_{1,1} - r_{1,2} + 2r_{1,3} \tag{3.35c}$$

$$\frac{\mathrm{d}C_{DEB_1}}{\mathrm{d}t} = \frac{-F_{out_1} C_{DEB_1}}{V_1} + r_{1,2} - r_{1,3} \tag{3.35d}$$

$$\frac{\mathrm{d}T_1}{\mathrm{d}t} = \frac{T_{1_o} F_1 - T_1 F_{out_1}}{V_1} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_1 C_p} r_{1,j} + \frac{Q_1}{\rho_1 C_p V_1} \tag{3.35e}$$

The dynamic model of the second CSTR is represented by the following ODEs:

$$\frac{\mathrm{d}C_{E_2}}{\mathrm{d}t} = \frac{F_2 C_{Eo_2} + F_{out_1} C_{E_1} - F_{out_2} C_{E_2}}{V_2} - r_{2,1} - r_{2,2} \tag{3.36a}$$

$$\frac{\mathrm{d}C_{B_2}}{\mathrm{d}t} = \frac{F_2 C_{Bo_2} + F_{out_1} C_{B_1} - F_{out_2} C_{B_2}}{V_2} - r_{2,1} - r_{2,3} \tag{3.36b}$$

$$\frac{\mathrm{d}C_{EB_2}}{\mathrm{d}t} = \frac{F_{out_1} C_{EB_1} - F_{out_2} C_{EB_2}}{V_2} + r_{2,1} - r_{2,2} + 2r_{2,3} \tag{3.36c}$$

$$\frac{\mathrm{d}C_{DEB_2}}{\mathrm{d}t} = \frac{F_{out_1} C_{DEB_1} - F_{out_2} C_{DEB_2}}{V_2} + r_{2,2} - r_{2,3} \tag{3.36d}$$

$$\frac{\mathrm{d}T_2}{\mathrm{d}t} = \frac{T_{2_o} F_2 + T_1 F_{out_1} - T_2 F_{out_2}}{V_2} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_2 C_p} r_{2,j} + \frac{Q_2}{\rho_2 C_p V_2} \tag{3.36e}$$

where the reaction rates are calculated by the following expressions:

$$r_{i,1} = k_1 \mathrm{e}^{\frac{-E_1}{RT_i}} C_{E_i} C_{B_i} \tag{3.37a}$$

$$r_{i,2} = k_2 \mathrm{e}^{\frac{-E_2}{RT_i}} C_{E_i} C_{EB_i} \qquad i = 1, 2 \text{ (reactor index)} \tag{3.37b}$$

$$r_{i,3} = k_3 \mathrm{e}^{\frac{-E_3}{RT_i}} C_{DEB_i} C_{B_i} \tag{3.37c}$$

Parameter values and steady-state values of the first-principles-based dynamic model are mentioned in Table 2.1.

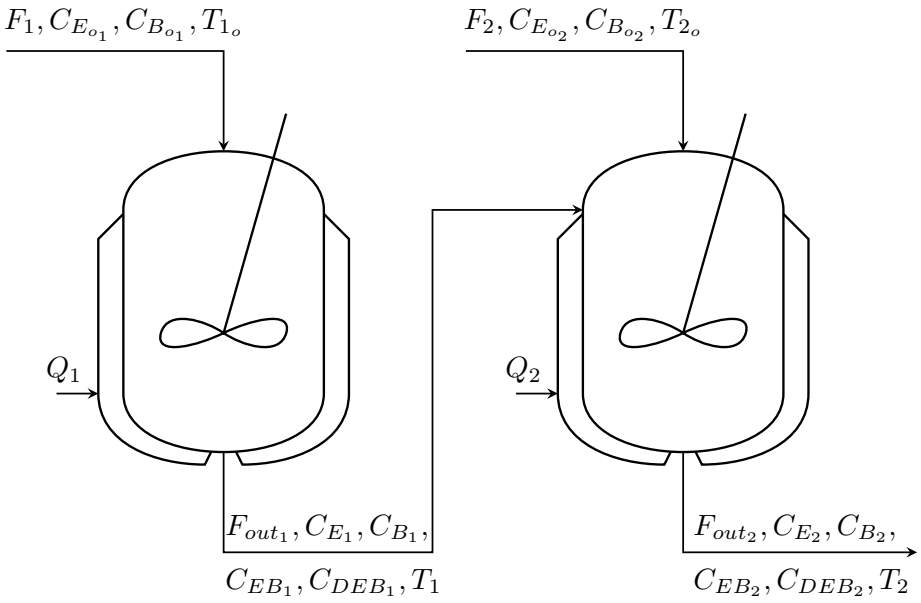

Figure 3.3: Process schematic featuring two CSTRs connected in series.

3.5.2 Implementing encryption in the two-tier control architecture

Prior to integrating encryption-decryption into a process, the selection of parameters, namely d, l_1, and l_2 is performed. An evaluation of the extreme feasible states and inputs guides the derivation of the integer bit count, denoted as $l_1 - d$. The upper limit in the $\mathbb{Q}_{l_1,d}$ set is obtained via the formula

$2^{l_1-d-1} - 2^{-d}$, whereas the lower limit is -2^{l_1-d-1}. The choice of the quantization parameter d rests on the desired accuracy and range of state and input values. Additionally, l_2 is chosen to exceed l_1. In alignment with this methodology, for the CSTR studied in this section, $l_1 - d$ is calculated to be 16, from which l_1 and d are then fixed. Within the set $\mathbb{Q}_{l_1,d}$, rational numbers are separated by a resolution of 2^{-d}, indicating that higher d values result in lower quantization errors. For simulation purposes, we opt for $d = 8$ as it yields nearly identical closed-loop state trajectories in comparison to the unencrypted case [39, 81]. These cited works also illustrate how the choice of the quantization parameter impacts closed-loop performance and stability across different values of d. For $d = 8$, we obtain $l_1 = 24$ and, since it is imperative that $l_2 > l_1$ for the subsequent bijective mapping, l_2 is selected as 30. With the quantization parameters defined, the next step entails the quantization of states and inputs, followed by their encryption via the Paillier Encryption algorithm. The implementation of the Paillier Encryption procedure is done through Python's "phe" module, PythonPaillier [21].

Remark 3.4. *As mentioned earlier, the implementation of encryption requires quantization of real-number valued signals to a fixed dataset denoted as $\mathbb{Q}_{l_1,d}$. The selection of quantization parameter $d = 8$ is justified by its enhanced control performance in comparison to lower values of d. The time needed for encryption computation can be divided into five distinct components, the time spent for: quantization of real data ($g_{l_1,d}$), bijective mapping ($f_{l_2,d}$), encryption, decryption, and inverse mapping ($f^{-1}_{l_2,d}$). [39] underscored that the encryption phase, followed by decryption, accounts for the majority of the time spent. Moreover, the time remains unaffected by the chosen quantization parameter. The three remaining mathematical operations contribute only a minimal portion of*

the total time spent on encryption-decryption. While the time taken for these operations does increase with quantization, the increment is insignificant compared to the total time spent, and the advantage of improved control performance using a higher quantization parameter greatly outweighs the slight increase in time. Hence, a quantization parameter of $d = 8$ *is adopted across all cases where encryption is implemented in this study.*

Implementation of the encrypted lower-tier control system

In the lower tier, control input computations are confined to linear mathematical operations, ensuring their execution within an encrypted space that guarantees cyber-security. The selection of lower-tier controlled inputs, which possess the capability to stabilize the entire system, is a pivotal task that requires adherence to a well-defined procedure. The procedure includes linearization of the nonlinear dynamical model about its operating steady-state, yielding a 10-dimensional state space model mirroring the number of states, governed by two control inputs—the heat removal rate for each CSTR. A, B, C, and D matrices were created for the state-space model $\dot{x} = \mathrm{A}x + \mathrm{B}u$ and $y = \mathrm{C}x + \mathrm{D}u$, where y are the observed measurements from the system. Subsequently, leveraging the Cohen-Coon tuning method, the control input gains are calibrated and further refined through multiple simulations conducted on the nonlinear dynamical model. Subsequently, the integral terms are omitted, substituting only proportional terms, $u = \mathrm{K}x$ in the state-space model, resulting in $\dot{x} = \mathrm{A}x + \mathrm{BK}x$. The eigenvalues of $(\mathrm{A} + \mathrm{BK})$ are then computed and verified to exhibit negative real components. This ensures asymptotic stability for the controllers when applied to the linearized model over the operating steady state. The inclusion of the integral term serves to eradicate offsets, thereby contributing to the refinement of closed-loop performance. Although

excluded in the eigenvalue computations, the integral terms were meticulously adjusted through a series of simulations using the nonlinear dynamical model. Next, via extensive simulations of the nonlinear system under the lower-tier controller, the controller is verified to adhere to the criteria outlined in Eq. (3.12), confirming that it can exponentially stabilize the system within the two-tier encrypted control framework.

Implementation of the encrypted LMPC in the upper-tier control system

The first-principles model, expressed by Eq. (3.35), serves as the foundational process model within the LMPC framework. For solving the constrained nonlinear, non-convex optimization problem, we leverage the Python module of the IPOPT software [83]. Consequently, the resultant solution is a local optimum, not a global one. This is a limitation due to the nature of the optimization that a global optimum cannot be found for such a problem [11]. The process of solving this optimization problem involves defining constraints for the LMPC. IPOPT constructs a feasibility region and employs an iterative methodology to progressively navigate towards the optimal solution by traversing the interior of the feasibility region. This approach incorporates two key parameters: the maximum number of iterations and a validation error. These parameters function as the stopping criteria within the optimization problem. If either of these conditions is met, IPOPT employs the final computed value as the solution for the given instance. Conversely, if neither of these criteria is satisfied, IPOPT reports the suboptimal values calculated in the last iteration, but the LMPC utilizes the control input calculated by the backup controller.

To assess the cost function of the LMPC over the prediction horizon, the integration step h_c is determined as $10^{-2} \times \Delta$ using the first-principles model. The positive definite matrix P in the con-

trol Lyapunov function $V = x^\top Px$ is selected as $\mathrm{diag}\ [250\ \ 500\ \ 500\ \ 1000\ \ 0.3\ \ 250\ \ 250\ \ 500\ \ 1000\ \ 0.6]$, drawing from extensive simulations. The LMPC framework employs a prediction horizon of $N = 2$. The stability criterion is defined as $\rho = 100$. Additionally, the criterion $\rho_{\min} = 1$ is the smaller level set of the Lyapunov function where the state is desired to be trapped. The weight matrices Q_1 and Q_2 in the LMPC cost function are chosen as $Q_1 = \mathrm{diag}\ [2000\ \ 2000\ \ 5000\ \ 5\ \ 5\ \ 2000\ \ 2000\ \ 5000\ \ 2\ \ 2]$ and $Q_2 = \mathrm{diag}\ [1\ \ 1\ \ 6\ \ 8]$, respectively. The cost function is defined as $L(x, u_{t2}) = x^\top Q_1 x + u_{t2}^\top Q_2 u_{t2}$.

Sampling time criteria with encryption

To implement encryption within a practical context, it is essential to ensure that the sampling time, Δ, surpasses the combined maximum duration required for encryption and decryption of all states and control inputs. Furthermore, it should accommodate the maximum time necessary for computing control actions at each sampling instance across the considered quantization parameter (d). This condition holds true for both the upper- and lower-tier control systems within an encrypted two-tier control framework. Mathematically,

$$\Delta_i > \max\left(\text{Encryption-decryption time}\right)_i + \max\left(\text{Control input computation time}\right)_i \tag{3.38}$$

where $i = \{1, 2\}$, with $i = 1$ and $i = 2$ representing the lower and upper control tier, respectively. In the discussed example, the sampling time Δ is chosen as 30 seconds. This decision is made while taking into account the previously mentioned requirement to implement the encryption process. Eq. (3.38) does not include the communication time required for signal transmission. This

is because the two-tier encrypted control architecture, discussed within the context of SCADA systems, relies on networked communication, which is extremely efficient and rapid. However, networked communication also exposes the system to cyberattacks, which is a vulnerability that we aim to mitigate in this research by introducing encryption to these communication channels.

Remark 3.5. *In the context of the discussed two-tier encrypted control architecture, the lower and upper tiers operate independently, maintaining distinct public keys for encryption and private keys for decryption. Consequently, they possess the flexibility to adopt different sampling times. For the CSTR example studied in this work, both tiers maintain identical sampling times. If certain control inputs necessitate shorter sampling periods and more frequent actuation, it is advisable to allocate them to the lower tier. The lower tier is a set of linear controllers and, hence, can compute control inputs more rapidly than an advanced nonlinear control scheme such as MPC employed in the upper tier. Also, as the lower tier does not perform encryption and decryption within the networked communication channels, it has less stringent sampling time constraints. Furthermore, strategies employing two-tier control to address challenges posed by delayed and asynchronous signals have been demonstrated in prior studies of [50, 54]. For systems incorporating delayed and asynchronous signals, these signals can be transmitted to the upper tier while applying control inputs through a sample-and-hold procedure. However, the primary motivation behind the adoption of the two-tier design in this research is the cyber-vulnerability of the upper tier due to the need to compute nonlinear control inputs without the safeguard of an encrypted computational environment.*

3.5.3 Cyberattack detector training and testing

In the upper-tier control system, the cyberattacks take the form of data manipulation. The objective involves crafting a detector capable of recognizing cyberattacks based on familiar data manipulation patterns as well as those it has not encountered previously. To accomplish this, a feed-forward neural network (FNN) is used to identify cyberattacks. The FNN is trained with min-max, replay, sinusoidal, and false-data-injection attacks. The FNN underwent testing with the aforementioned attacks, along with the inclusion of surge and geometric attacks. The outcome of the FNN is categorized into two classes: "cyberattack" and "no attack". Each data point in the dataset represents a 1×40 array of $V(x)$ values. To instill variability, we employed a spectrum of initial conditions, mirroring a range of process scenarios. The activation of an attack was randomly timed between $i_o \in [5, 35]$ to create diverse durations and occurrences during system operation. Throughout the training phase, a randomized approach was adopted, wherein an attack would be simulated on a single state measurement for each CSTR at random intervals. In the testing phase, a similar random approach was followed, wherein cyberattacks were introduced on either one or multiple state measurements or control inputs.

To build the training and validation set, we conducted extensive closed-loop simulations, resulting in a dataset comprising 6000 data points. Each class ("cyberattack" and "no attack") contained 3000 data points. For the cyberattack class, 750 data points per attack type were included in the training. The dataset was divided into an 80:20 ratio for training and validation purposes. Employing feature-wise normalization prevented overfitting and enhanced results. For the testing phase, a separate set of 1200 data points was generated – 600 for each class and 100

data points for each cyberattack type. To account for real-world process fluctuations and avoid mistaking minor variations as cyberattacks, bounded Gaussian white noise was incorporated into each sensor measurement, for all the data points. By bounding the noise, the tail ends of the Gaussian distributed noise are eliminated before being applied. The cited work of [77] proposes methods to deal with tail-ends in Gaussian-distributed noise. The sensor noises were constrained within the following bounds: $|\omega_i| \leq 0.1$, $\forall i = \{1, 2, 3, 6, 7, 8\}$; $|\omega_i| \leq 0.0003$, $\forall i = \{4, 9\}$; $|\omega_i| \leq 0.35$, $\forall i = \{5, 10\}$; these Gaussian noise distributions have zero mean and standard deviations $|\sigma_i| \leq 0.03$, $\forall i = \{1, 2, 3, 6, 7, 8\}$; $|\sigma_i| \leq 0.0001$, $\forall i = \{4, 9\}$; $|\sigma_i| \leq 0.1$, $\forall i = \{5, 10\}$. In this context, the subscripts are associated with different system states. Subscripts 1, 2, 3, 4, and 5 denote the concentrations of ethylene, benzene, ethylbenzene, di-ethyl benzene, and reactor temperature for CSTR 1, respectively. Similarly, subscripts 6, 7, 8, 9, and 10 correspond to the concentrations of ethylene, benzene, ethylbenzene, di-ethyl benzene, and reactor temperature for CSTR 2, respectively.

The design of the feed-forward neural network structure followed a systematic approach. It comprised 40 input neurons, each corresponding to normalized control Lyapunov function values derived from the previous 40 sampling instances. The FNN was designed with two hidden layers, while the output neurons were set to 2, aligning with the binary classification task at hand. Fixing the number of neurons in the hidden layers, selecting the optimizer, and specifying activation functions before the hidden layers was established through a meticulous grid search process. The number of epochs was fixed to 100 during the grid search. The objective was to identify the optimal combination of hyperparameters that yielded the lowest validation loss. Based on the results, the configuration of the neural network architecture included 60 neurons in the first hidden layer

and 25 neurons in the second hidden layer. To mitigate the risk of overfitting, a dropout ratio of 0.2 was applied after each hidden layer. The activation functions employed were as follows: hyperbolic tangent after the input layer, rectified linear unit (ReLU) after the first hidden layer, and softmax after the second hidden layer. Upon tuning with these hyperparameters, the model underwent 1000 epochs of training using the Adam optimizer with the objective to minimize the sparse categorical cross-entropy loss. Throughout the training, emphasis was placed on conserving the model configuration that exhibited the lowest validation loss. This meticulous approach facilitated the development of an effective and well-optimized neural network model for subsequent testing and evaluation. The training, validation, and testing accuracies for the model are 99.87%, 99.92%, and 99.83%, respectively.

Remark 3.6. *As outlined in Section* 3.5.3*, the cyberattack is introduced randomly within the sampling instances ranging from [5, 35], covering a span of 40 instances from which data is gathered for the control Lyapunov function for a single data point. Attacks launched after sampling instance 30 pose a relatively higher challenge for cyberattack detection algorithms. Within the following sampling instances, these attacks may not induce substantial deviations in the process dynamics. This is due to the model being trained with noisy data to prevent ordinary process fluctuations from being misidentified as cyberattacks. However, as the attacks persist and gradually push the system away from the desired stability region* $\Omega_{\rho_{\min}}$ *(but still within* Ω_{ρ}*), their detectability becomes more feasible. Hence, while the accuracies might not reach 100%, practical implementation within a system reveals the potential to achieve cyberattack detection with near-perfect accuracy and very slightly extended response times.*

Remark 3.7. *Di-ethyl benzene is an unintended byproduct that emerges within the reaction scheme elucidated in Eq.* (3.34)*. It exists in minimal quantities within both CSTRs and is not a direct control input. Consequently, in the process of randomly initiating cyberattacks on the state values received by the MPC for training, validation, and testing datasets, no cyberattacks are launched on the state values of di-ethyl benzene. This omission stems from the recognition that cyberattacks on state values of di-ethyl benzene would exert no discernible influence on the overall process dynamics. For this reason, cyberattacks are exclusively aimed at the eight other state values received by the MPC, as well as all four control inputs computed by the MPC in the upper-tier control system. Given its trace presence, visual depictions of its concentration are not included in this section. However, di-ethyl benzene is considered as a system state for the purpose of process modeling and MPC calculations. Consequently, all results presented account for its presence within the system.*

3.5.4 Two-tier control architecture without cyberattack detection and reconfiguration mechanisms

In this section, we illustrate the two-tier control architecture without incorporating any detection and control reconfiguration mechanism. Figure 3.4 visually illustrates all six discussed cyberattacks. The cyberattacks are launched at time $t = 0.5\,\mathrm{hr}$. The true state measurements of the concentration of ethylene in CSTR 1 of the process network, depicted by the solid green line, stand in contrast to the manipulated state values received by the MPC during a cyberattack. The altered values, indicated by the dashed red line, emerge due to the manipulation of the true state values received by the MPC during the cyberattack. Evidently, the actual state values and the received values by the MPC diverge in opposite directions as the actuation is executed based on the

received values, rather than the authentic state values.

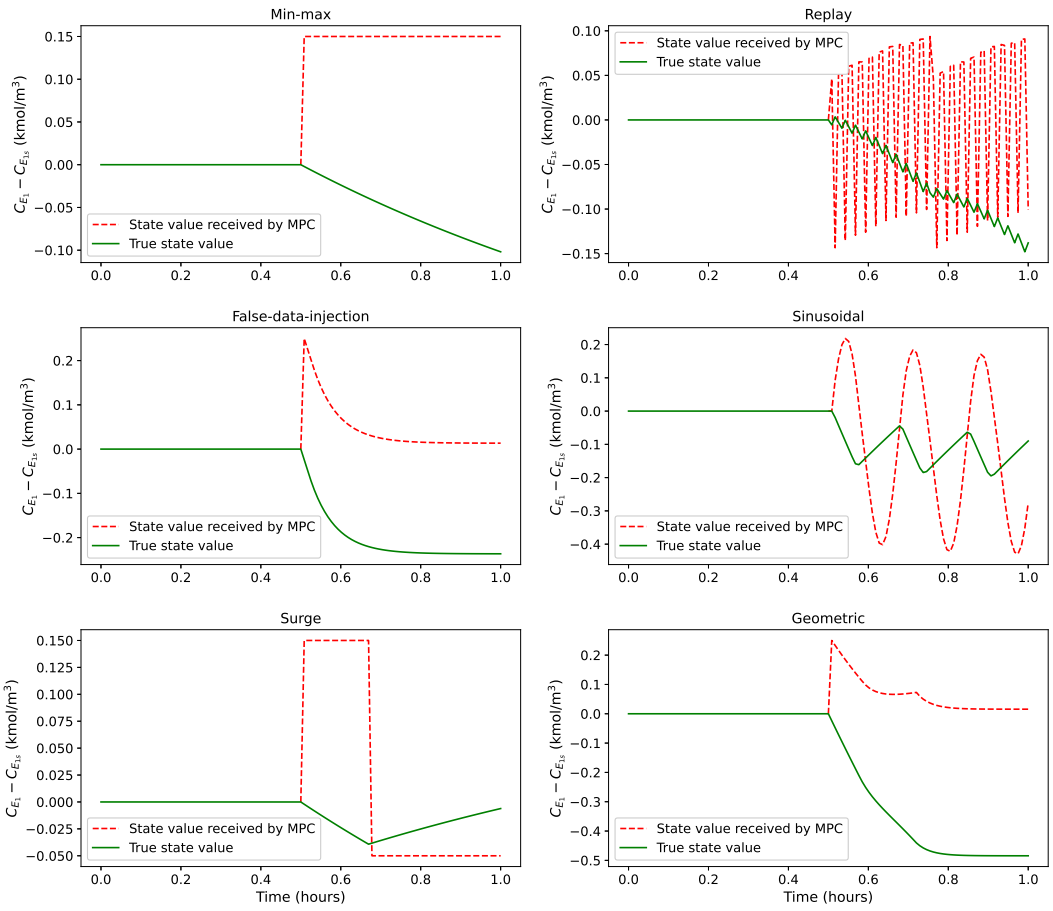

Figure 3.4: True state value of $C_{E_1} - C_{E_{1s}}$ (green solid line) and state value of $C_{E_1} - C_{E_{1s}}$ received by the MPC (red dashed line) for all the cyberattacks discussed.

To portray the overall impact of a cyberattack on the system in the absence of a detection mechanism, a geometric cyberattack is executed on two state measurements of the upper tier at time $t = 0.5\,\mathrm{hr}$. The attack targets the state values associated with the concentration of ethylene in CSTR 1 and the concentration of benzene in CSTR 2 that are received by the MPC. As evident

from Figures 3.5 and 3.6, the cyberattack does not destabilize the system beyond the stability region Ω_ρ. The final value of the control Lyapunov function $V(x)$ at $t = 2\,\mathrm{hr}$ is 33.65 which is within the stability limit, $\rho = 100$. Nevertheless, it does lead to a continued reduction in the concentration of ethyl benzene in CSTR 2—the desired product—resulting in economic loss. The lower tier, responsible for controlling the heat inputs to both CSTRs and is fully safeguarded against cyberattacks, prevents attacks on the upper tier from completely destabilizing the system. However, the integration of a machine learning-based cyberattack detection mechanism can deactivate the upper tier, thereby ensuring system stabilization within the desired stability region $\Omega_{\rho_{\min}}$. Furthermore, conventional detection mechanisms based on fail-safe boundary conditions, like identifying an attack when the value of the control Lyapunov function surpasses $\rho = 100$, would prove inadequate in detecting an intelligent cyberattack.

3.5.5 Simulation results of the encrypted two-tier control architecture with cyberattack detection and re-configuration mechanisms

In this section, we employ the encrypted two-tier control architecture, featuring a machine learning-based cyberattack detector and a reconfiguration mechanism to disable the upper tier upon cyberattack detection. Two distinct scenarios are presented: one where the system operates at an unstable steady-state and the other where the system is converging to an unstable steady-state while remaining within the stability region Ω_ρ. The objective of intelligent cyberattacks is to inflict harm on the process yield without causing the system to exit the stability region. As a result, we do not delve into cyberattacks launched when the system states are outside Ω_ρ, as conventional detection mechanisms are sufficient for addressing such cases.

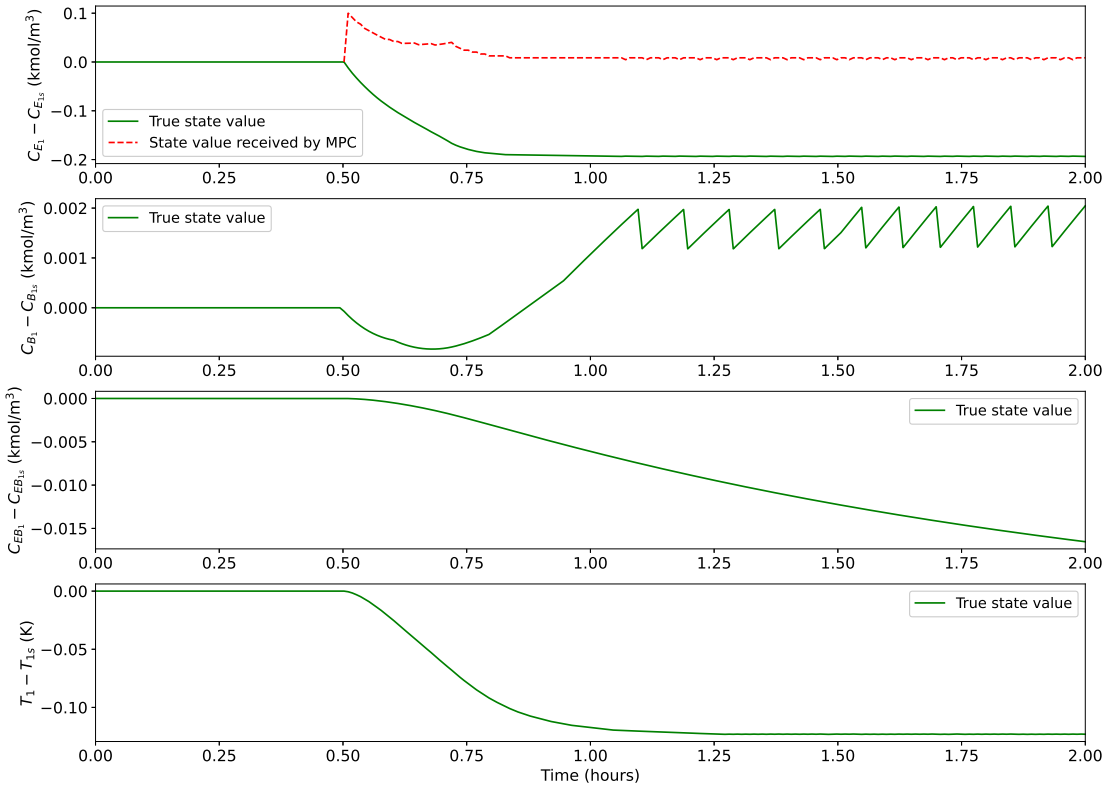

Figure 3.5: True state values (green solid line) and state value received by the MPC (red dashed line) for CSTR 1 without detection and reconfiguration mechanisms when a geometric cyberattack is launched at $t = 0.5\,\text{hr}$.

In both scenarios under consideration, the cyberattack is initiated at $t = 0.383$ hours or 23 minutes of process time. As mentioned, the upper-tier control inputs are the inlet concentration of ethylene and benzene for each CSTR and the lower-tier control inputs are the heat removal rates for each CSTR. Figures 3.7 to 3.9 depict the first scenario, where the control inputs computed by the MPC before encryption are manipulated via a surge attack when the system is operating at its unstable steady-state. Figures 3.10 to 3.12 depict the second scenario, where the state values of the system received by the MPC after decryption are manipulated via a geometric attack when the system is converging to its unstable steady state. In all the figures in Section 3.5.5, the operating control scheme is illustrated through different colored lines. The red line depicts the system under

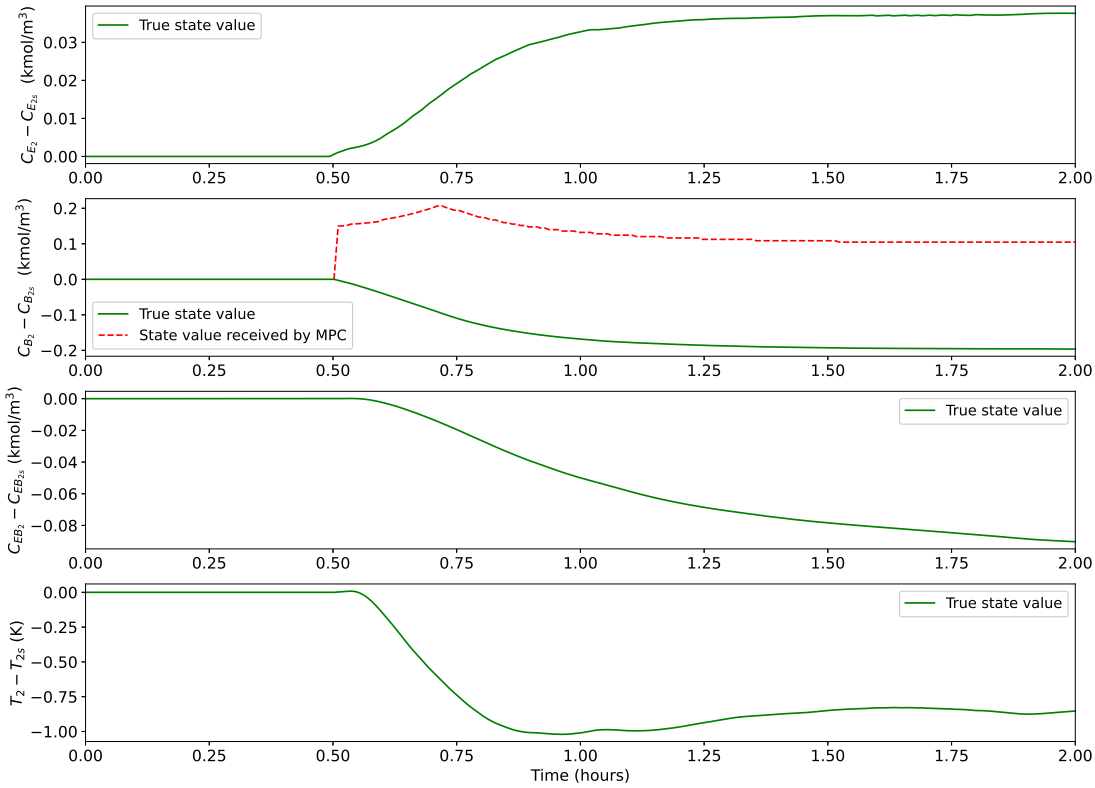

Figure 3.6: True state values (green solid line) and state value received by the MPC (red dashed line) for CSTR 2 without detection and reconfiguration mechanisms when a geometric cyberattack is launched at $t = 0.5\,\text{hr}$.

the two-tier encrypted control scheme, the red line marked with stars depicts the system under the two-tier encrypted control scheme during a cyberattack, and the green line depicts the system under solely the lower-tier control scheme after the cyberattack has been detected, and the system has been reconfigured. It is worth noting that the ML-based cyberattack detector was not trained on the geometric and surge attack patterns. Yet, the detector demonstrated its ability to promptly identify these attacks.

In Figures 3.7 to 3.9, during the initial 23 minutes of the process time, flat trajectories are observed for all the states and control inputs as the system is operating at its unstable steady-state under the two-tier encrypted control scheme without any cyberattack. At $t = 23\,\text{min}$, a

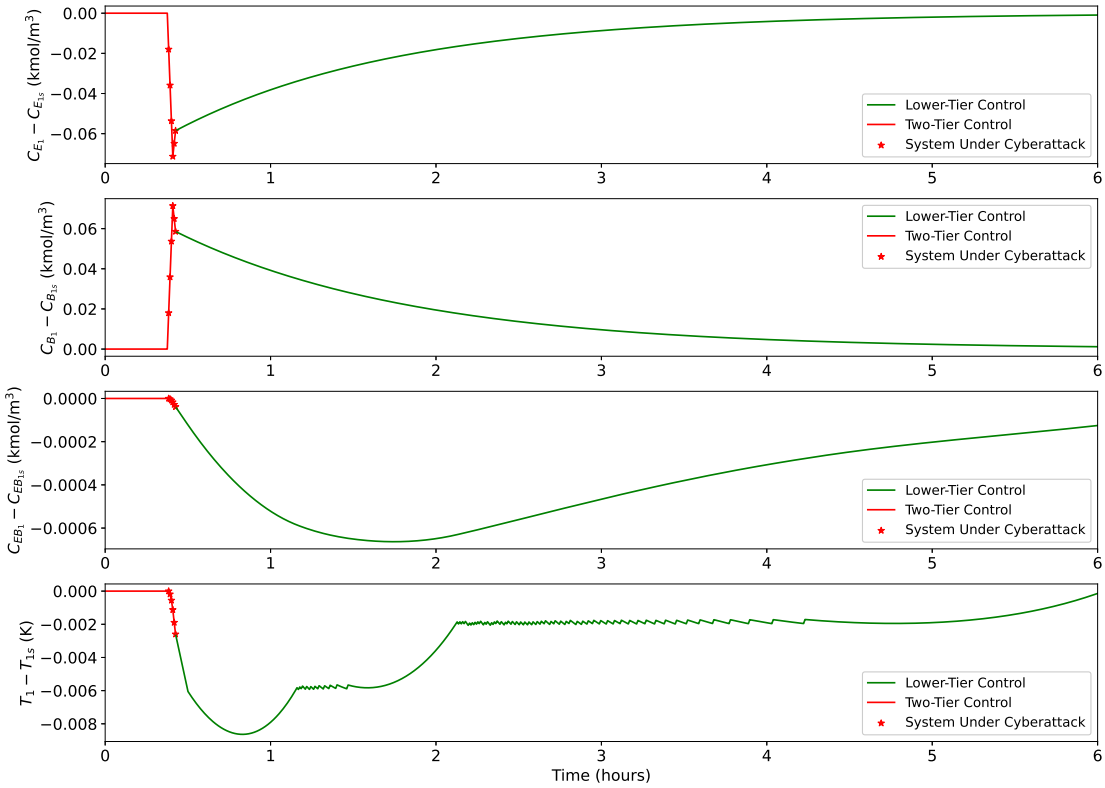

Figure 3.7: State profiles of CSTR 1 under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a surge attack is launched on the upper-tier control inputs at $t = 23\,\text{min}$.

surge cyberattack is launched on all four control inputs of the upper tier by manipulating the MPC control inputs before they are encrypted. This manipulation deviates the system from its desired stability region, $\Omega_{\rho_{\min}}$, without complete destabilization. The cyberattack detector begins detecting the attack status at each sampling instance after 20 minutes, requiring data from the preceding 40 sampling instances (equivalent to 20 minutes of process time). The detector identifies the cyberattack for the first time at 25 minutes, 2 minutes after the attack was initiated on the upper tier control inputs. After three consecutive detections at 25, 25.5, and 26 minutes, the upper tier control is disabled at 26 minutes. Subsequently, only the secure, encrypted lower-tier control scheme is employed to guide the system back to its desired stability region, $\Omega_{\rho_{\min}}$.

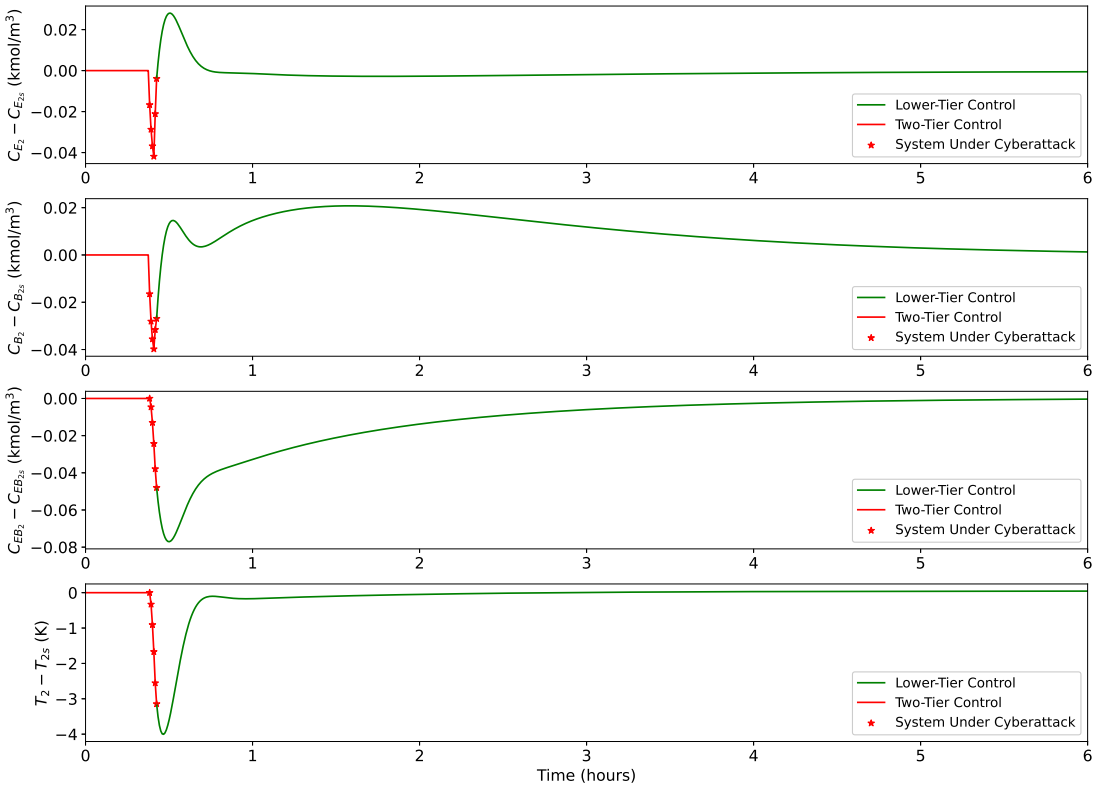

Figure 3.8: State profiles of CSTR 2 under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a surge attack is launched on the upper-tier control inputs at $t = 23\,\text{min}$.

For Figures 3.10 to 3.12, during the initial 23 minutes of the process time, the state trajectories exhibit swift convergence towards their steady-states as they operate under the two-tier encrypted control scheme without any cyberattack. At $t = 23\,\text{min}$, a geometric cyberattack is initiated, targeting all 6 concentration states of the upper tier. This attack involves manipulating the decrypted state values received by the MPC in their plaintext form, and it deviates the system states from their prior converging trajectory towards their steady-states. The cyberattack detection mechanism commences after 20 minutes of the process, necessitating data from the preceding 40 sampling instances (equivalent to 20 minutes). The cyberattack detector first identifies the cyberattack at the 26 minutes, 3 minutes after the cyberattack was initiated. After three consecutive detection

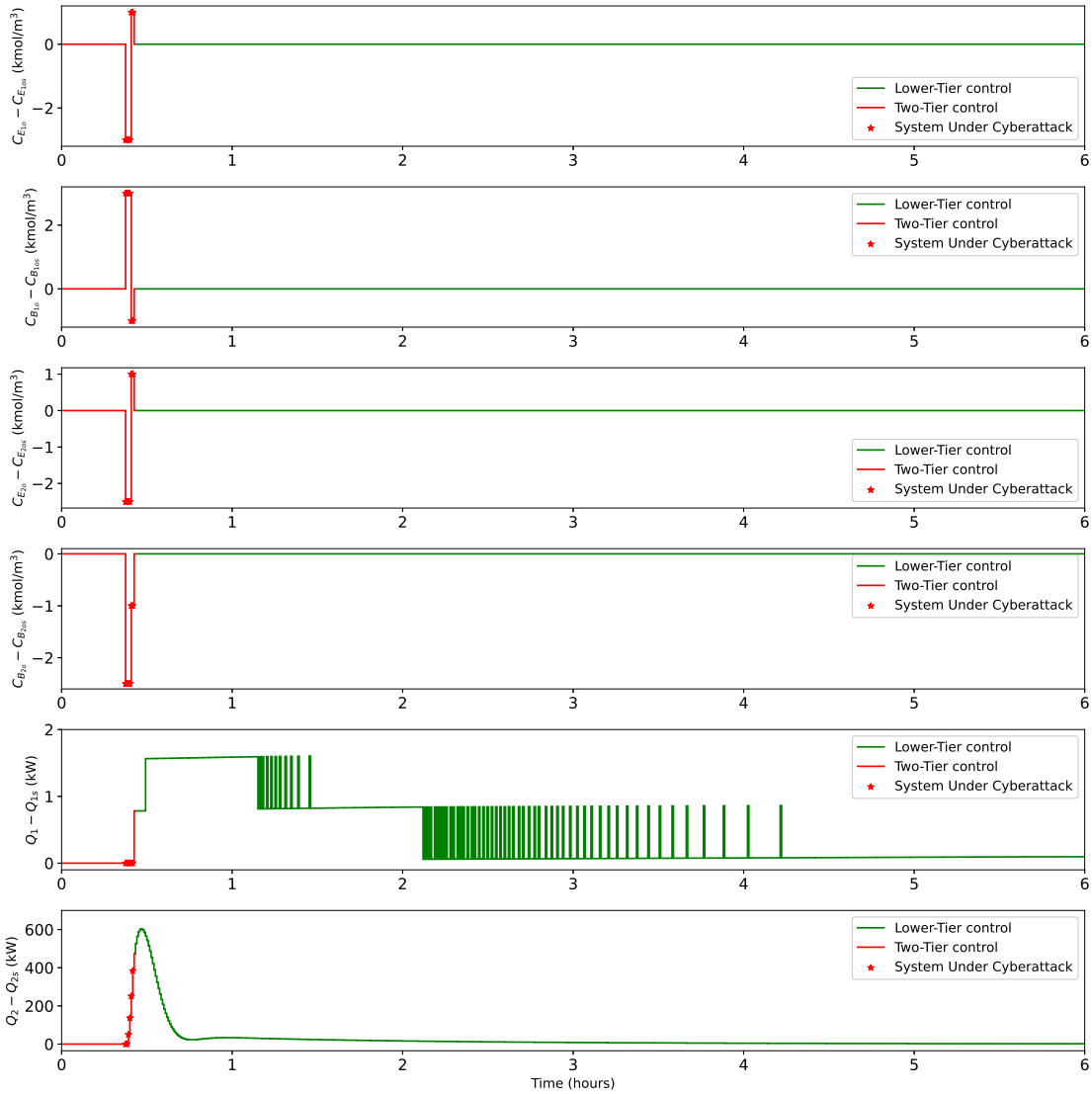

Figure 3.9: Control input profiles under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a surge attack is launched on the upper-tier control inputs at $t = 23\,\mathrm{min}$.

instances at 26, 26.5, and 27 minutes, the upper-tier control scheme is disabled at 27 minutes. Subsequently, only the secure, encrypted lower-tier control scheme is employed to guide the system back to its desired stability region, $\Omega_{\rho_{\min}}$. Also, in this scenario, when the cyberattack is launched, the system is in the process of converging towards its steady state; it has not yet reached its final equilibrium. Importantly, the cyberattack detector remains active and can effectively identify

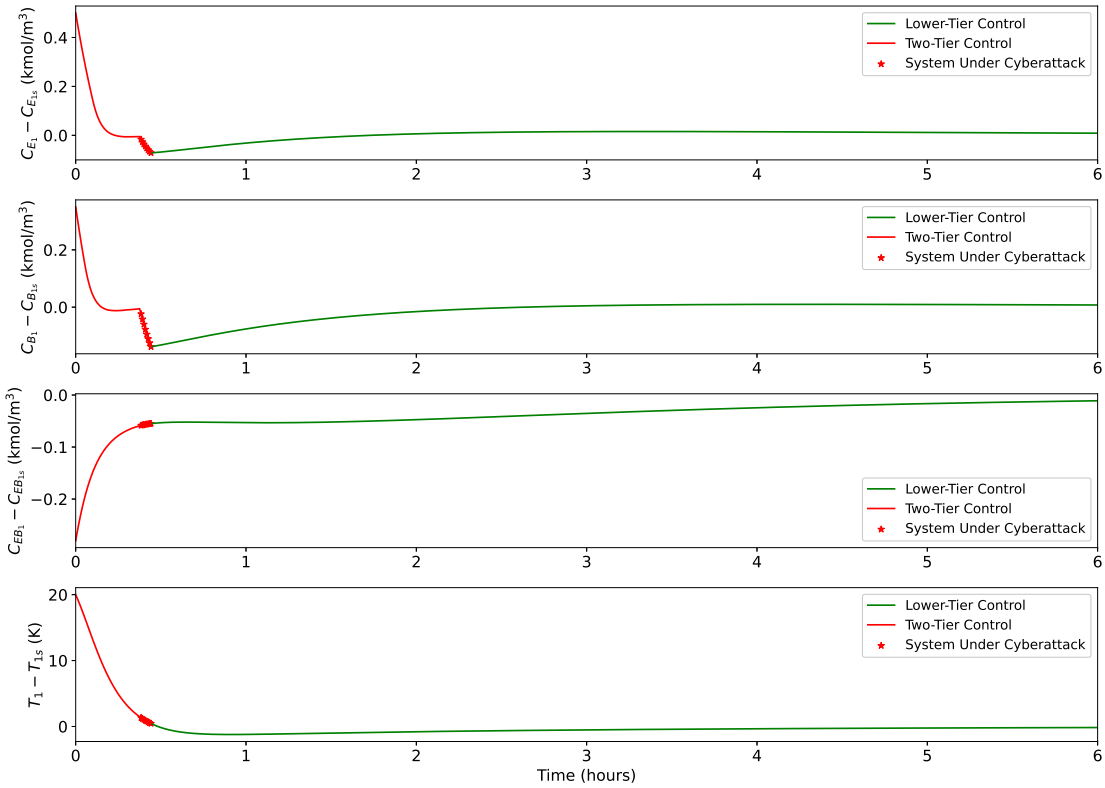

Figure 3.10: State profiles of CSTR 1 under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a geometric attack is launched on the upper-tier states at $t = 23\,\text{min}$.

attacks even during this transitional phase, as illustrated inFigures 3.10 to 3.12.

Remark 3.8. *The presence of quantization introduces some irregularities in the curves of certain control inputs and states. For example, in Figure 3.9, noticeable bumps can be observed in the control input response corresponding to the rate of heat removal in CSTR 1. These bumps stem from the fact that the quantization error value is multiplied by the gains of the controller within an encrypted framework. As a result, these multiplicative effects generate bumps in the trajectories of control inputs. However, in the case of the rate of heat removal for CSTR 2, this phenomenon is not as apparent in the same figure. This is attributed to the significantly larger magnitude of the control input for CSTR 2. Similarly, this irregularity is not observed in the inlet concentration*

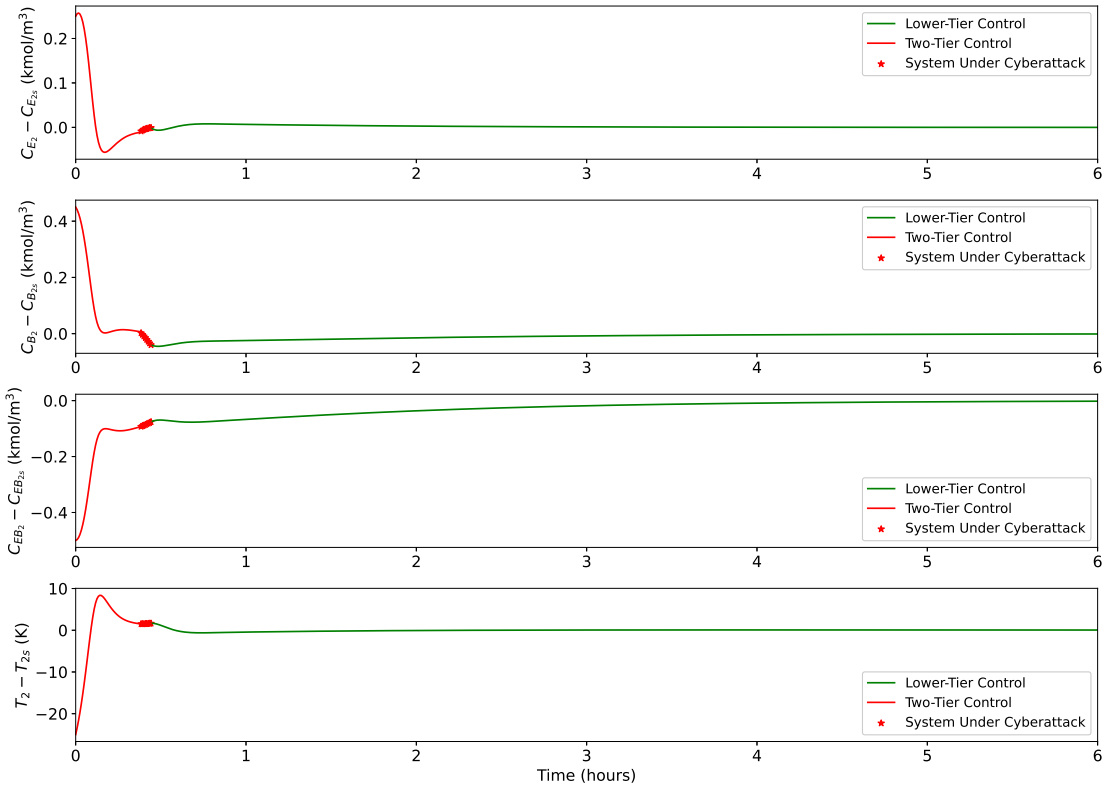

Figure 3.11: State profiles of CSTR 2 under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a geometric attack is launched on the upper-tier states at $t = 23\,\mathrm{min}$.

control inputs for the CSTRs, as these control inputs are quantized after the plain text computation by the MPC. This approach prevents the multiplication of quantized terms, thus mitigating the generation of bumps due to control input quantization. Although the quantization effects are less conspicuous in the case of inlet concentration control inputs, their discontinuous behavior resulting from quantized terms still exists. This effect is mitigated by selecting a higher quantization parameter. As a solution, a quantization parameter of $d = 8$ has been opted for all the simulations that are being presented. This choice of a higher quantization parameter helps alleviate the observed irregularities.

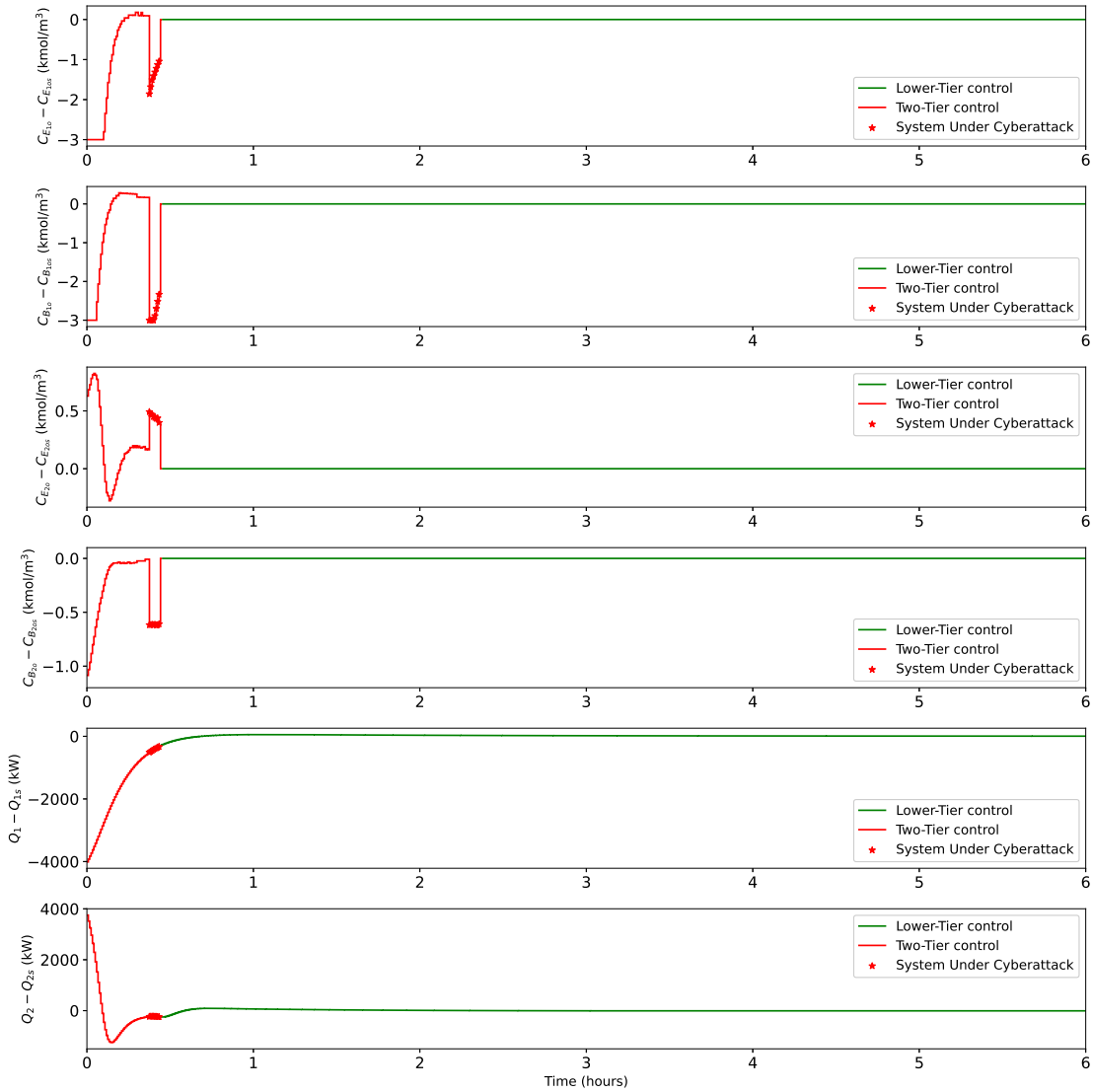

Figure 3.12: Control input profiles under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a geometric attack is launched on the upper-tier states at $t = 23\,\mathrm{min}$.

3.5.6 Computational time of ML-based detection compared to encryption-decryption

This subsection delves into the computational load implications of incorporating machine-learning-based cyberattack detection within the encrypted control framework. A comparative analysis

was conducted between the time dedicated to cyberattack detection and the time allocated to the encryption-decryption of upper-tier states and control inputs. The lower tier is fully secure as it maintains encrypted communication throughout the network (sensor–controller–actuator), thus rendering detection algorithms unnecessary for it. Due to the independent operation of the lower tier in relation to the upper tier, along with the redundancy of cyberattack detection for the lower tier, the time taken for encryption in it is excluded from this analysis.

Figure 3.13 depicts the ratio of the time taken for cyberattack detection to the time required for encryption-decryption operations for 25 minutes of process time, corresponding to 50 consecutive sampling instances. It is evident from Figure 3.13 that the ML-based cyberattack detection consumes, on average, less than 1% of the time required for encryption-decryption. Consequently, the integration of this detection mechanism does not impose a significant computational burden on the overall time complexity of the system. Instead, it introduces a crucial cybersecurity aspect, especially in situations where the encrypted upper tier might not be entirely cyber-secure due to the context in which plaintext data encryption or decryption occurs within the control architecture.

Remark 3.9. *In this chapter, the lower tier of the two-tier encrypted control architecture functions as a secure, stabilizing feature in continuous operation throughout the process. When a cyberattack is detected, only the upper-tier is deactivated, while the lower tier continues to stabilize the system without any interruptions. Alternatively, in a different framework than the one proposed in this research, the lower tier controller can serve as a backup controller within the architecture if it is desired for the MPC to exclusively compute all control inputs. In such a scenario, when an attack is detected, control would be transitioned from the upper tier to the lower tier, which remains inactive*

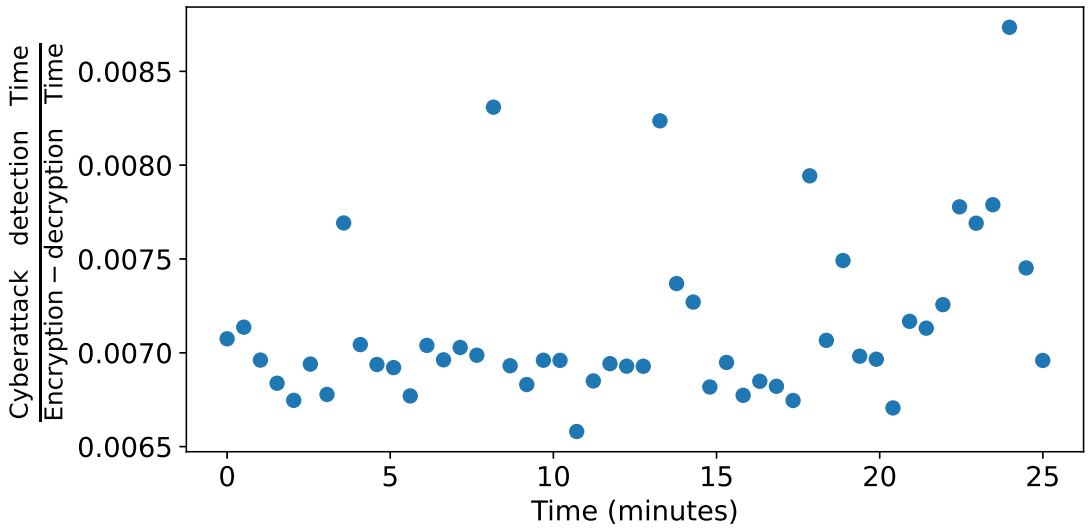

Figure 3.13: Ratio of the time for ML-based Cyberattack detection to encryption-decryption for 50 consecutive sampling periods.

during normal process operation when no cyberattack is detected. Thus, at any time only one tier would be functional. Nevertheless, as previously stated, in this study, both the lower and upper tiers remain operational in the absence of any cyberattack. In the event of a detected cyberattack, the upper tier is deactivated, while the lower tier continues its role in stabilizing the system.

3.6 Conclusions

In this chapter, we presented an encrypted two-tier control architecture incorporating an ML-based cyberattack detector to enhance the operational safety, cybersecurity, and closed-loop performance of nonlinear process systems. The lower-tier control system comprises a set of encrypted proportional-integral controllers, while the upper-tier control system employs an encrypted Lyapunov-based model predictive controller. This architecture enhances system cybersecurity, even in settings where control input computations may not be cybersecure. By integrating both

linear and nonlinear controllers with encryption, the developed two-tier control architecture can be adapted to large-scale nonlinear processes. Further, we have provided insights into the framework and formulation of the encrypted lower- and upper-tier control systems. Through a comprehensive stability analysis, we have identified potential sources of error and established bounds to ensure closed-loop system stability. Additionally, we have delved into the development of an ML-based cyberattack detector, addressed critical aspects such as quantization parameter selection, sampling time criteria, and computational load assessment. These issues are essential for the practical implementation of the proposed control system across nonlinear processes. To validate the efficacy of our control framework, we subjected it to previously unseen cyberattack patterns within a nonlinear chemical process network utilized in ethylbenzene production. We carried out a detailed simulation study that exposed the implementation and performance of the two-tier control architecture and the usefulness of the cyberattack detector. In summary, our work advances control system cybersecurity by integrating ML-based cyberattack detection into encrypted control systems with both linear and nonlinear controllers.

Chapter 4

Encrypted Decentralized Model Predictive Control of Nonlinear Processes with Delays

4.1 Introduction

Numerous large-scale industrial systems, such as power distribution grids, mechanical systems, chemical processes, and urban traffic networks, present a significant challenge as the system to be controlled is too large, resulting in a complex control problem to be solved. This challenge cannot be simply solved by using faster computers with larger memory. In response, decentralized control strategies have been proposed to address high dimensionality, constraints related to information structure, and inherent system delays in such systems [7]. In a decentralized setup, the overall system is divided into independent sub-systems that may be coupled with each other, but controlled by separate controllers, which together constitute a decentralized control structure. This provides a practical solution for reducing the computational complexity of a centralized control problem for a large-scale process.

Alongside dealing with large-scale processes, it is crucial to address the various sources of time delays that can impact control systems. These sources include the computation of control inputs for large-scale systems, communication lags during signal transfer, the inherent dynamics of material transportation within the process system, and control actuator dynamics. Using a decentralized control structure reduces a large, complex control problem into smaller sub-problems which are solved independently and simultaneously in different computing units. This reduces the delays due to control input computation. Advances in networked communication have simplified the interlinking, connectivity, and data transfer in cyber-physical systems and has made time-delays from communication negligible. However, delays due to control actuator dynamics in process networks cannot be compensated by smaller control input computation times or rapid transport of material in processes or rapid networked communication and, hence, need appropriate control strategies such as integrating a predictor within the controller design [79]. Similarly, state delays in process networks cannot be completely eliminated by optimizing process layouts, and, hence, need to be accounted in the controller design.

Networked communication might make data transfer seamless and rapid. However, they are prone to cyberthreats. A breach or compromise within these systems could result in severe consequences, such as the disruption of critical services, physical harm, financial loss, and are also a threat to public safety. Recent developments in cyberattack techniques underscore the need to establish robust cybersecurity [31]. Addressing cybersecurity concerns within industrial control systems primarily falls under the domain of operational technology (OT). Significant progress has been made in enhancing cybersecurity in the information technology (IT) sector, which focuses on the software aspects of systems, encompassing areas such as network architecture and data

management. However, the field of cybersecurity within OT is currently trailing behind [16]. Various real-world examples underscore the importance of cybersecurity in networked cyber–physical systems and SCADA (Supervisory Control and Data Acquisition) environments. For instance, the cyberattack on SCADA controls responsible for managing the power grid in Ukraine in 2015 led to widespread power outages [45]. Similarly, in the DarkSide ransomware attack on Colonial Pipeline in 2021, hackers encrypted its networked communication and demanded a ransom for the decryption keys. Consequently, Colonial Pipeline had to halt operations, causing disruptions in fuel distribution and financial losses [82].

Extensive research has been conducted in areas such as developing machine learning-based cyberattack detectors [2, 26], using reachable set-based detection schemes [63], employing linear encrypted controllers [18, 20], analyzing the safety of process equipment when the system is under a cyberattack [65], control switching techniques for attack detection [62], process state recovery post cyberattack [87], and creating cyberattack-resilient controllers [68]. However, to the best of our knowledge, the development of cybersecure decentralized controllers for large-scale nonlinear processes with input and state delays remains an unexplored area, prompting our proposal for a novel control structure to address this challenge.

Specifically, we propose a decentralized control structure consisting of a set of Lyapunov-based MPCs, integrated with a predictor, utilizing encrypted networked communication. MPC is an advanced control strategy that achieves superior performance compared to traditional controllers via constraints, and optimizes critical performance metrics in multi-input multi-output systems. These advantages are derived from the utilization of a nonlinear mathematical model to predict future system behavior, and optimizing control inputs by minimizing a cost function with

constraints. For large-scale systems, the control problem to be solved by a centralized MPC would be too complex. In contrast, a decentralized MPC divides this intricate problem into smaller, independent segments, concurrently solved in different edge computing units. In this configuration, we assume the presence of secure edge computers responsible for computing control inputs. Integrating a predictor within this setup serves to offset performance degradation due to input delays. To mitigate the influence of state-delays resulting from material transportation in systems, the process model employed by the LMPCs in the decentralized framework is based on differential difference equations. These equations account for the inherent state-delays present in the system. Further, the incorporation of encryption within the networked communication channels enhances cybersecurity as each edge computing unit receives and transmits encrypted signals.

The remainder of the paper is organized as follows: Section 4.2 presents preliminaries on notation, the general class of nonlinear systems considered, the system stabilizability assumptions, the cryptosystem used for employing encryption, and the effect of quantization. The encrypted decentralized MPC design, formulation of the MPCs, and stability analysis of the control scheme are presented in Section 4.3. In Section 4.4, closed-loop simulations of a nonlinear chemical process network with input and state delays under the encrypted decentralized LMPC system with and without predictor feedback are presented and discussed.

4.2 Preliminaries

4.2.1 Notation

The symbol $\|\cdot\|$ represents the Euclidean norm of a vector. $x^\top$ denotes the transpose of a vector x. $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ represent the sets of real numbers, integers, and natural numbers, respectively. $\mathbb{Z}_M$ denotes the additive groups of integers modulo M. Set subtraction is indicated by the symbol "\", where $A\backslash B$ represents the set of elements that are in set A but not in set B. A function, $f(\cdot)$, falls under the class $\mathcal{C}^1$ if it is continuously differentiable within its defined domain. The term $\mathrm{lcm}(i,j)$ denotes the least common multiple of the integers i and j, while $\gcd(i,j)$ signifies the greatest common divisor, that divides i and j without any remainder.

4.2.2 Class of systems

This research focuses on multi-input multi-output (MIMO) nonlinear time-delay systems, characterized by a set of differential difference equations (DDEs), alternatively known as delay differential equations. These equations are formulated in the following manner:

$$\dot{x} = F(x,u) = f(x(t), x(t-d_1), u(t-d_2)) \tag{4.1}$$

The state vector is denoted by $x \in \mathbb{R}^n$, while $u \in \mathbb{R}^m$ represents the control input vector bounded by the set, $U \subset \mathbb{R}^m$. $d_1 > 0$ and $d_2 > 0$ are the state and input delays, respectively. The vector $f(\cdot)$ is a locally Lipschitz vector function of its arguments with $f(0,0,0) = 0$, rendering the origin as a steady state of Eq. (4.1). Without loss of generality, we assume the initial time as zero ($t_0 = 0$). Additionally, we define the set $S(\Delta)$ as the set of piece-wise constant functions characterized by a

period of Δ. We consider $j = 1, \ldots, N_{sys}$ sub-systems, with each subsystem j consisting of states x_j which are regulated only by inputs u_j but potentially impacted by states in other subsystems due to coupling between subsystems. The continuous-time nonlinear dynamics of subsystem j is described as follows:

$$\dot{x}_j = F_j(x, u_j), \quad x_j(t_0) = x_{j_0}, \quad \forall j = 1, \ldots, N_{sys} \tag{4.2}$$

where N_{sys} denotes the number of subsystems, $x_j \in \mathbb{R}^{n_j}$ and $u_j \in \mathbb{R}^{m_j}$ are the state vector and control inputs for subsystem j, respectively. $x = [x_1^\top \ldots x_{N_{sys}}^\top]^\top \in \mathbb{R}^n$ is the state vector for the entire system, with $n = \sum_{j=1}^{N_{sys}} n_j$. $u = [u_1^\top \ldots u_{N_{sys}}^\top]^\top \in \mathbb{R}^m$ is the control input vector for the entire system, with $m = \sum_{j=1}^{N_{sys}} m_j$. The control input vector constraints are $u_j \in U_j := \{u_{\min,j_i} \leq u_{j_i} \leq u_{\max,j_i} \, \forall i = 1, 2, \ldots, m_j\} \in \mathbb{R}^{m_j}$, $\forall j = 1, \ldots, N_{sys}$. Hence, the set U that constrains the control input vector for the entire system is formed by the union of sets U_j, where $j = 1, \ldots, N_{sys}$. The system of Eq. (4.1) can be expressed as a perturbed form of the system without delays in the following manner:

$$\dot{x} = F(x, u, \xi) = f(x(t), x(t) + \xi_1(t), u(t) + \xi_2(t)) \tag{4.3a}$$

$$\xi_1 = x(t - d_1) - x(t) \tag{4.3b}$$

$$\xi_2 = u(t - d_2) - u(t) \tag{4.3c}$$

where $\xi^\top := [\xi_1^\top, \xi_2^\top] \in D \times U \in \mathbb{R}^{n+m}$ is the bounded perturbation vector, and D is the open neighborhood around the origin.

Remark 4.1. *In this research, we employ differential difference equations to characterize nonlinear time-delay systems. Differential difference equations (DDEs) fundamentally differ from ordinary differential equations (ODEs). One key distinction is that a dynamic system with an arbitrarily small delay is considered an infinite-dimensional system, even though the state vector would have finite dimension. Existing literature offers various approaches to describe nonlinear time-delay systems, such as first-order plus dead time and second-order plus dead time models. However, these methods are specific and assume certain linear model structures. Hence, we have opted to utilize nonlinear differential difference equations with constant delays in this study to ensure a more comprehensive analysis. Nevertheless, it is worth noting that other studies have utilized functional differential equations to describe nonlinear time-delay systems [34], and our findings can potentially be extended to encompass such model structures as well.*

4.2.3 Stability assumptions

Based on how the overall large-scale system is partitioned, there may exist interacting dynamics between the subsystems, as the states of one subsystem may impact the states of other subsystems. Accounting for these interactions, we assume the existence of stabilizing control laws $u_j = \Phi_j(x) \in U_j$, which regulate the individual subsystems $j = 1, \ldots, N_{sys}$, such that the origin of the overall system of Eq. (4.1) with $d_1 \equiv 0$ and $d_2 \equiv 0$ is rendered exponentially stable. This signifies the presence of a $\mathcal{C}^1$ control Lyapunov function $V(x)$ for which the following inequalities hold for all $x \in \mathbb{R}^n$ within an open region D surrounding the origin:

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2, \tag{4.4a}$$

$$\frac{\partial V(x)}{\partial x} f(x, x, \Phi(x)) \leq -c_3|x|^2, \tag{4.4b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \leq c_4|x| \tag{4.4c}$$

where c_1, c_2, c_3 and c_4 are positive constants. $\Phi(x) = [\Phi_1(x)^\top \dots \Phi_{N_{sys}}(x)^\top]^\top$ is the vector concatenating the stabilizing feedback control laws for all N_{sys} subsystems. For the nonlinear system described by Eq. (4.1), the region of closed-loop stability can be defined as a level set, Ω_ρ, of the control Lyapunov function V, such that $\Omega_\rho := \{x \in D | V(x) \leq \rho\}$, where $\rho > 0$. Hence, originating from any initial condition within Ω_ρ, the control input, $\Phi(x)$, guarantees that the state trajectory of the closed-loop system remains within Ω_ρ.

4.2.4 Paillier cryptosystem

In this research, we employ the Paillier cryptosystem [67] to encrypt signals, specifically state measurements (x) and control inputs (u), transmitted to and from the controllers. Although we do not make use of the semi-homomorphic property of additive homomorphism within the Paillier cryptosystem, we employ it so that traditional controllers, such as proportional-integral controllers, which conduct computations in an encrypted space, can be integrated into the overall control architecture if required. The encryption procedure is initiated by generating the public and private key. The public key is used to encrypt integer messages into ciphertexts, and the private key is employed to decrypt ciphertexts and retrieve the original integer messages. The process of generating the public and private key can be outlined as follows:

1. Choose two large prime integers (p and q) randomly, ensuring, $\gcd(pq, (p-1)(q-1)) = 1$.

2. Compute, $M = pq$.

3. Choose an arbitrary integer $\bar{g}$ such that $\bar{g} \in \mathbb{Z}_{M^2}$,which is the multiplicative group of integers modulo M^2.

4. Compute $\lambda = \mathrm{lcm}(q-1, p-1)$.

5. Specify $\bar{L}(x) = (x-1)/M$.

6. Verify the existence of the subsequent modular multiplicative inverse:
   $u = (\bar{L}(\bar{g}^{\lambda} \bmod M^2))^{-1} \bmod M$.

7. If the inverse does not exist, revisit step 3 and select an alternate value of $\bar{g}$. If the inverse exists, $(M, \bar{g})$ is the public key and (λ, u) is the private key.

Once the keys are acquired, the public and private keys are distributed to authorized recipients for encryption and decryption, respectively. The encryption process is as follows:

$$E_M(m, r) = c = \bar{g}^m r^M \bmod M^2 \tag{4.5}$$

where r is a randomly selected integer from the set $\mathbb{Z}_M$, and c represents the ciphertext achieved through the encryption of m. The decryption procedure is as follows:

$$D_M(c) = m = \bar{L}(c^{\lambda} \bmod M^2)u \bmod M \tag{4.6}$$

The aforementioned procedure can be demonstrated in a numerical example as follows:

Key generation steps:

1. Select 2 prime numbers $p = 13$, and $q = 17$.

2. $M = p \times q = 13 \times 17 = 221$.

3. Chose, $\bar{g} = 8$ which can be any integer between 1 and M^2.

4. Calculate $\lambda = \text{lcm}(q-1, p-1) = \text{lcm}(16, 12) = 48$.

5. Verify the existence of $u = 172$.

6. The public key is $(M, \bar{g}) = (221, 8)$.

7. The private key is $(\lambda, u) = (48, 172)$.

Encryption:

1. The message to be encrypted is $m = 3$.

2. A random number $r = 1$ is chosen such that $0 < r < M$.

3. The ciphertext is: $c = \bar{g}^m r^M \mod M^2 = 8^3 1^{221} \mod 221^2 = 512$.

Decryption:

1. The ciphertext to be decrypted is $c = 512$.

2. The message is $m = \bar{L}(c^\lambda \bmod M^2)u \bmod M = \bar{L}(512^{48} \bmod 221^2)172 \mod 221 = 3$.

4.2.5 Quantization

To use the Paillier cryptosystem, data to be encrypted must be in the form of natural numbers in $\mathbb{Z}_M$. However, the signal values before encryption are in floating-point. Consequently, we employ

quantization, mapping the floating-point numbers into $\mathbb{Z}_M$ [19]. Using a signed fixed-point binary representation, we create a set, $\mathbb{Q}_{l_1,d}$, with parameters l_1 and d. These parameters define the total bit count (integer and fractional) and the fractional bits, respectively. The $\mathbb{Q}_{l_1,d}$ set encompasses rational numbers from -2^{l_1-d-1} to $2^{l_1-d-1} - 2^{-d}$, separated by 2^{-d}. A rational number q in $\mathbb{Q}_{l_1,d}$ can be expressed as $q \in \mathbb{Q}_{l_1,d}$, where $\exists \beta \in \{0,1\}^{l_1}$, and $q = -2^{l_1-d-1}\beta_{l_1} + \sum_{i=1}^{l_1-1} 2^{i-d-1}\beta_i$. To map a real number data point a to the $\mathbb{Q}_{l_1,d}$ set, we use the function $g_{l_1,d}$, defined by the equation,

$$\begin{aligned} g_{l_1,d} &: \mathbb{R} \to \mathbb{Q}_{l_1,d} \\ g_{l_1,d}(a) &:= \arg\min_{q \in \mathbb{Q}_{l_1,d}} |a - q| \end{aligned} \tag{4.7}$$

Next, the quantized data is transformed into a set of integers through a one-to-one (bijective) mapping known as $f_{l_2,d}$, as outlined in [19]. The following mapping ensures that the quantized data is transformed into a subset of the message space $\mathbb{Z}_M$:

$$\begin{aligned} f_{l_2,d} &: \mathbb{Q}_{l_1,d} \to \mathbb{Z}_{2^{l_2}} \\ f_{l_2,d}(q) &:= 2^d q \bmod 2^{l_2} \end{aligned} \tag{4.8}$$

During the encryption process, integer plaintext messages from the set $Z_{2^{l_2}}$ are converted to ciphertexts, which can be decrypted back into the same set $Z_{2^{l_2}}$. To recover the original data from the set $\mathbb{Q}_{l_1,d}$, an inverse mapping, denoted as $f^{-1}_{l_2,d}$, is defined as follows:

$$f^{-1}_{l_2,d} : \mathbb{Z}_{2^{l_2}} \to \mathbb{Q}_{l_1,d} \tag{4.9}$$

$$f_{l_2,d}^{-1}(m) := \frac{1}{2^d} \begin{cases} m - 2^{l_2} & \text{if } m \geq 2^{l_2-1} \\ m & \text{otherwise} \end{cases} \tag{4.10}$$

4.3 Development of the encrypted decentralized control architecture

In this section, we describe the design of the encrypted decentralized control architecture, establish bounds on the errors in the encrypted decentralized control structure through a stability analysis, and formulate the predictor feedback-based decentralized LMPC.

4.3.1 Design of the encrypted decentralized control architecture

In the encrypted decentralized control architecture, depicted in Figure 4.1, at time t_k, where k represents the sampling instance, signals $x(t_k)$ from sensors are encrypted to ciphertext c using the public key and transmitted to each control subsystem, within its respective edge computing unit. Within each unit, the encrypted signals are decrypted using the private key, and the quantized states $\hat{x}(t_k)$ are used to initialize the predictor in the j^{th} control subsystem, where j ranges from 1 to N_{sys}. The predictor computes the states after the input delay period, $\hat{x}(t_k + d_2)$. This is used to initialize the nonlinear process model of the j^{th} MPC. Subsequently, the j^{th} MPC computes the optimized control input trajectory along the whole prediction horizon and encrypts the control input $u_j(t_k + d_2)$. At the actuator, the ciphertext $\acute{c}$ is decrypted to the quantized input $\hat{u}(t_k + d_2)$. However, due to the input delay, d_2, the control input applied to the process by the actuator is $\hat{u}(t_k)$, which was calculated at time $t_k - d_2$. Since the data received and transmitted by the edge

computers through the network remains encrypted, cybersecurity is ensured in the presence of secure edge computers.

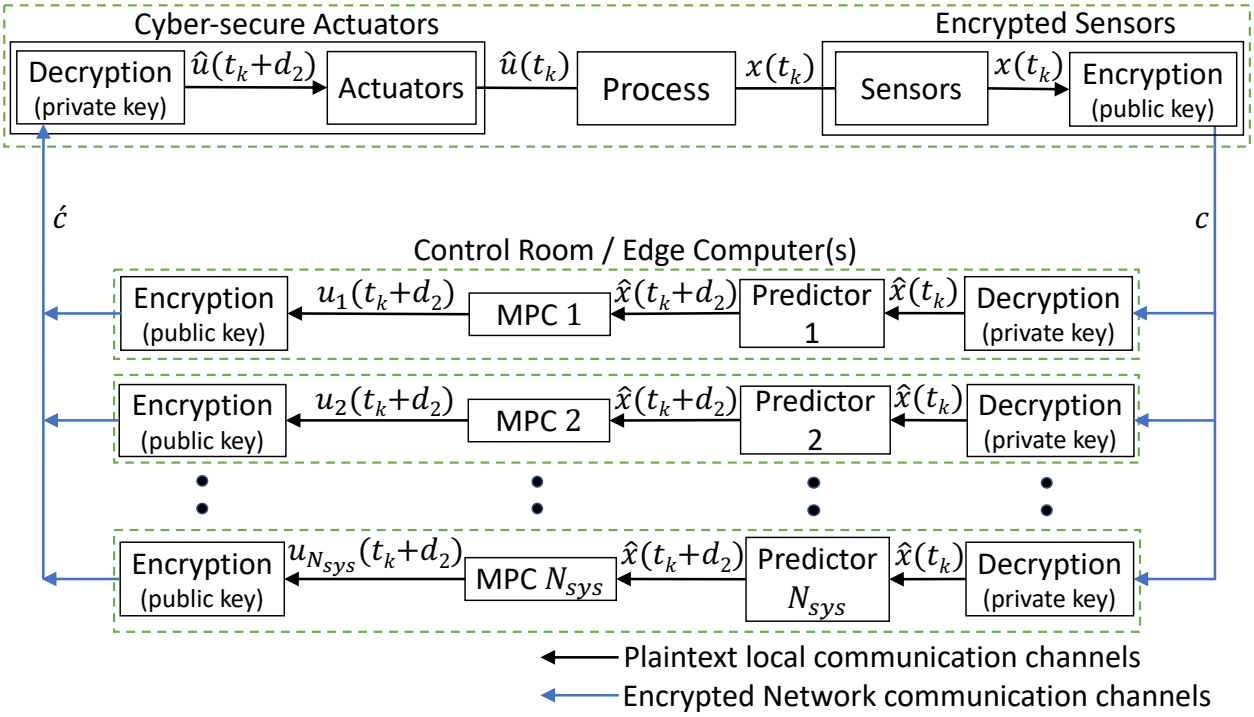

Figure 4.1: Illustration of the encrypted decentralized control structure.

The closed-loop design of Figure 4.1 introduces two sources of error: one from state quantization in the sensor–controller link and another from control input quantization in the controller–actuator link. These errors are bounded by:

$$|x(t_k) - \hat{x}(t_k)| \leq 2^{-d-1} \tag{4.11a}$$

$$|u(t_k) - \hat{u}(t_k)| \leq 2^{-d-1} \tag{4.11b}$$

The derivation of the upper bounds of the quantization error in Eq. (4.11) has been explained in Remark 4.3. An additional error arises in the applied control input as the predictor, $\phi(x, u)$,

receives $\hat{x}$ instead of the true state x to predict the states after the input delay period. Using the local Lipschitz property, this error will be confined by the underlying equation, where $L_1' > 0$:

$$|\phi(\hat{x}, u) - \phi(x, u)| \leq L_1'|\hat{x} - x| \leq L_1' 2^{-d-1} \tag{4.12}$$

Remark 4.2. *In this work, a decentralized MPC, without inter-controller communication, is proposed to reduce the computational time and complexity of a centralized control problem. For possibly superior performance, some level of communication between controllers in different subsystems may be necessary to account for coupling effects between subsystems in large-scale processes. To establish this, a distributed control architecture could be used. However, encrypting-decrypting control input trajectories multiple times within a single sampling period could significantly increase the communication overhead due to encryption. To avoid this, a secure Ethernet crossover cable connection could be established between different computing units in a single control room responsible for computing all the control inputs of a particular process. This would avoid the need for encrypting-decrypting control inputs as their transmission would be secure, and encryption could still be used for signals transmitted to and from the control room.*

Remark 4.3. *Quantization error arises when the value to be quantized is not found exactly in the set $\mathbb{Q}_{l_1,d}$. The elements in this set are separated by 2^{-d}. Let us assume the value to be quantized is a, which lies between b and $b + 2^{-d}$. If the absolute difference between a and b is less than that between a and $b + 2^{-d}$, a is mapped to b, while, otherwise, a is mapped to $b + 2^{-d}$. Thus, the maximum potential difference between the actual value and the quantized value is half of the resolution or 2^{-d-1}. Further, this bound implies that a higher value of d would result in a smaller*

error due to quantization.

4.3.2 Decentralized LMPC

To reduce the computational time and complexity of a centralized control problem, we formulate a decentralized LMPC system as follows:

$$\mathcal{J}_j = \min_{u_{d_j} \in S(\Delta)} \int_{t_k}^{t_{k+N}} L_j(\tilde{x}_j(t), u_{d_j}(t))\, \mathrm{d}t \tag{4.13a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}_j(t) = F_j(\tilde{x}(t), u_{d_j}(t)) \tag{4.13b}$$

$$u_{d_j}(t) \in U_j,\ \forall\, t \in [t_k, t_{k+N}) \tag{4.13c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{4.13d}$$

$$\dot{V}(\hat{x}(t_k), u_{d_j}(t_k)) \leq \dot{V}(\hat{x}(t_k), \Phi_j(\hat{x}(t_k))),$$

$$\text{if } \hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{4.13e}$$

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

$$\text{if } \hat{x}(t_k) \in \Omega_{\rho_{\min}} \tag{4.13f}$$

Each LMPC has access to full-state feedback measurements but only takes into account the dynamics of its respective subsystem. Consequently, we develop separate first-principles-based models for each subsystem j where $j = 1, \ldots, N_{sys}$, to predict the states x_j and compute the control input u_{d_j} to be applied by the corresponding actuators within the j^{th} subsystem. $\tilde{x}_j$ represents the predicted state trajectory of the process model of the j^{th} LMPC. The quantized states, $\hat{x}$, serve as the initial conditions for the LMPC process model to predict the state trajectory as per Eq. (4.13b),

which is used to integrate the cost function of Eq. (4.13a) to calculate optimized control inputs, $u^*_{d_j}(t|t_k)$, for the entire prediction horizon. However, the LMPC transmits only the first input of this sequence to the actuator for application to the system within the interval $t \in [t_k, t_{k+1})$ and repeats this process at each sampling period. k is the sampling instance, and N represents the number of sampling periods within the prediction horizon. Eq. (4.13c) represents the constraints imposed on the control inputs, and Eq. (4.13d) uses the quantized states to initialize the plant model described in Eq. (4.13b). The Lyapunov constraint in Eq. (4.13e) ensures that, if the state $x(t_k)$ at time t_k lies within the set $\Omega_\rho \setminus \Omega_{\rho_{\min}}$, where $\rho_{\min}$ represents a level set of V in proximity to the origin, the time-derivative of the control Lyapunov function of the closed-loop subsystem j under the j^{th} LMPC is less than or equal to the time-derivative of the control Lyapunov function when the subsystem is controlled by the stabilizing controller $\Phi_j(x)$. When the closed-loop state $x(t_k)$ enters $\Omega_{\rho_{\min}}$, the constraint of Eq. (4.13f) ensures that this state remains within $\Omega_{\rho_{\min}}$.

4.3.3 Robustness of the encrypted decentralized LMPC to time-delay systems

In this subsection, we will focus on the closed-loop stability analysis of the perturbed nonlinear system of Eq. (4.3), taking into consideration sufficiently small state delays only (i.e., $d_2 \equiv 0$ and $d_1 > 0$). However, the stabilization of the perturbed system of Eq. (4.3) in the presence of both state and input delays will be achieved using an encrypted decentralized LMPC with predictor feedback in Section 4.3.4. We first establish stability of the closed-loop system under the encrypted stabilizing controller $\hat{\Phi}(\hat{x})$, followed by extending our analysis to stability of the system under the encrypted decentralized LMPC system introduced in the previous section.

Theorem 4.1. *Considering the system of Eq.* (4.3) *under the encrypted stabilizing controller* $\hat{\Phi}(\hat{x})$*, we examine the stability of the time-delay system without any input delay (i.e.,* $d_2 \equiv 0$ *and* $d_1 > 0$*). The stabilizing controller* $\Phi(x)$*, without encryption and delays, adheres to the inequalities outlined in Eq.* (4.4)*. Furthermore, we assume that the initial state* x_0 *resides within the region* $\Omega_{\hat{\rho}}$ *where* $\hat{\rho} < \rho$*. Given a sufficiently large time* $T > 0$*, where* T *is the time needed for* $x(t)$ *to enter* $\Omega_{\rho_{\min}}$*, we can determine positive real numbers* $L'_x, L'_\xi, L'_q, M_F, M_{d_1}$*, and* $e_t = (L_1 + 1)2^{-d-1}$*, for which there exist* Δ, d_1, d*, and* $\epsilon_w > 0$*, such that the following conditions are satisfied:*

$$\begin{aligned} &L'_x M_F \Delta + L'_\xi M_{d_1} d_1 + L'_q |e_t| - \frac{c_3}{c_2}\rho_s \leq -\epsilon_w \\ &\rho_{\min} = \max\{V(x(t+\Delta)) | V(x(t)) \leq \rho_s\} \end{aligned} \tag{4.14}$$

where $\hat{\rho} > \rho_{\min} > \rho_s$*. Then, the closed-loop state* $x(t)$ *under the encrypted stabilizing controller remains bounded in* $\Omega_{\hat{\rho}}$ *and is ultimately bounded in* $\Omega_{\rho_{\min}}$ *for* $t \geq T$*.*

Proof. This proof is divided into four parts. First, we will establish bounds on the error due to quantization in the time-delay system under the encrypted stabilizing controller, keeping the input delay, $d_2 \equiv 0$. Then, we will establish bounds for the error due to state delays, followed by limiting the error due to the control input being applied in a sample-and-hold manner. Lastly, based on these bounds, we can determine the positive constants $L'_x, L'_\xi, L'_q, M_F, M_{d_1}$, and $e_t = (L_1 + 1)2^{-d-1}$, for which there exist Δ, d_1, d, and $\epsilon_w > 0$, such that the state of the closed-loop system from any initial condition $x_0 \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ converges within $\Omega_{\rho_{\min}}$ within time T. Under the encrypted stabilizing controller, the control input $u(t)$ can be written as $u(t) = \hat{\Phi}(\hat{x}(t_k))$. Substituting this in the nonlinear system of Eq. (4.3) without any input delay (i.e. $d_2 \equiv 0$), the time-derivative of the

control Lyapunov function can be written as:

$$\dot{V} = \frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \hat{\Phi}(\hat{x}(t_k))) \tag{4.15}$$

Based on the error bounds resulting from quantization, as derived in Eq. (4.11), $\hat{\Phi}(\hat{x}(t_k)) \leq \Phi(\hat{x}(t_k)) + 2^{-d-1}$,

$$\begin{aligned} \dot{V} &\leq \frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \Phi(\hat{x}(t_k)) + 2^{-d-1}) \\ &\leq \frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \Phi(x(t_k)) + \Phi(\hat{x}(t_k)) - \Phi(x(t_k)) + 2^{-d-1}) \end{aligned} \tag{4.16}$$

Using the Lipschitz property, $\Phi(\hat{x}(t_k)) - \Phi(x(t_k)) \leq L_1|\hat{x} - x| \leq L_1 2^{-d-1}$. Substituting this in Eq. (4.16):

$$\begin{aligned} \dot{V} &\leq \frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \Phi(x(t_k)) + (L_1 + 1)2^{-d-1}) \\ &\leq \frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \Phi(x(t_k)) + e_t) \end{aligned} \tag{4.17}$$

where $e_t = (L_1 + 1)2^{-d-1}$ represents the error due to quantization. Using the constraints outlined in Eq. (4.4), Eq. (4.17) can be re-written as:

$$\begin{aligned} \dot{V} \leq\ & \frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \Phi(x(t_k)) + e_t) - \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), x(t_k), \Phi(x(t_k))) \\ & + \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), x(t_k), \Phi(x(t_k))) \\ \leq\ & \frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \Phi(x(t_k)) + e_t) - \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), x(t_k), \Phi(x(t_k))) \\ & - c_3|x(t_k)|^2 \end{aligned} \tag{4.18}$$

Based on Eq. (4.18), we can define the following: $g(x, \xi_1, e_t) = f(x, x+\xi_1, \Phi(x)+e_t)$. In addition, there exist positive constants, L'_x, L'_ξ, and L'_q such that the following Lipschitz inequality holds for

all $x, x' \in \Omega_{\hat{\rho}}$:

$$\left| \frac{\partial V(x)}{\partial x} g(x, \xi_1, e_t) - \frac{\partial V(x')}{\partial x} g(x', 0, 0) \right| \leq L'_x |x - x'| + L'_\xi |\xi_1| + L'_q |e_t| \tag{4.19}$$

Thus, Eq. (4.18) can be re-written as:

$$\begin{aligned} \dot{V} &\leq \frac{\partial V(x(t))}{\partial x} g(x(t), \xi_1(t), e_t) - \frac{\partial V(x(t_k))}{\partial x} g(x(t_k), 0, 0) - c_3 |x(t_k)|^2 \\ &\leq L'_x |x(t) - x(t_k)| + L'_\xi |\xi_1(t)| + L'_q |e_t| - c_3 |x(t_k)|^2 \end{aligned} \tag{4.20}$$

The upper bound of the perturbation term ξ_1 due to state delays can be represented as:

$$|\xi_1(t)| = |x(t - d_1) - x(t)| \leq d_1 M_{d_1} \tag{4.21}$$

where $M_{d_1} = \max_{s \in [-d_1, 0]} |x(t + s)|, \forall t \in [0, T]$. Substituting the bound on $|\xi_1(t)|$ derived from Eq. (4.21), we obtain:

$$\dot{V} \leq L'_x |x(t) - x(t_k)| + L'_\xi d_1 M_{d_1} + L'_q |e_t| - c_3 |x(t_k)|^2 \tag{4.22}$$

Due to the continuity of $x(t)$ $\forall\, t \in [t_k, t_k + \Delta)$, we can write that $|x(t) - x(t_k)| \leq M_F \Delta, \forall t \in [t_k, t_k + \Delta)$. Using this bound and the inequalities of Eq. (4.4), it follows from Eq. (4.22):

$$\dot{V} \leq L'_x M_F \Delta + L'_\xi d_1 M_{d_1} + L'_q |e_t| - \frac{c_3}{c_2} \rho_s \tag{4.23}$$

In the above equation, the first term represents the error due to sample-and-hold implementation of the control input, the second term represents the error due to state delays, and the third term

represents the error due to quantization. All these errors are bounded and can be made sufficiently small by constraining the sampling time and state delay to be sufficiently small, and using a higher quantization parameter d for encryption. Therefore, their sum is also bounded and can be made sufficiently small. This implies that, for the chosen time T, there exist positive real numbers Δ, d_1, d, and ϵ_w, such that the following inequality holds:

$$L_x' M_F \Delta + L_\xi' d_1 M_{d_1} + L_q' |e_t| - \frac{c_3}{c_2}\rho_s \leq -\epsilon_w, \forall t \in [0, T]$$

which implies that $\dot{V} \leq -\epsilon_w$ for any $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_s}$ for all $t_k \in [0, T]$. This establishes that, if the conditions of Eq. (4.14) are met, the closed-loop system state under the encrypted stabilizing controller is always bounded in $\Omega_{\hat{\rho}}$ and converges to $\Omega_{\rho_s} \subseteq \Omega_{\rho_{\min}}$ within time T, and remains there. □

Below, we proceed with the stability proof of the closed-loop system under the encrypted decentralized MPC.

Theorem 4.2. *Considering the system of Eq.* (4.3) *under the encrypted decentralized LMPC of Eq.* (4.13)*, we examine the stability of the time-delay system without any input delay (i.e.,* $d_2 \equiv 0$ *and* $d_1 > 0$*). We assume that the initial state* x_0 *resides within the region* $\Omega_{\hat{\rho}}$*. Given a sufficiently large time* $T > 0$*, where* T *is the time needed for* $x(t)$ *to enter* $\Omega_{\rho_{\min}}$*, we extend the results obtained in Theorem 4.1 to the encrypted decentralized LMPC of Eq.* (4.13) *maintaining our previous*

assumption that $\hat{\rho} > \rho_{\min} > \rho_s$. Then, if the following conditions are satisfied,

$$\begin{aligned} &\dot{V} \leq L_x' M_F \Delta + L_\xi' d_1 M_{d_1} + L_q' |e_t| - \frac{c_3}{c_2}\rho_s \leq -\epsilon_w \\ &\rho_{\min} = \max\{V(x(t+\Delta)) | V(x(t)) \leq \rho_s\} \end{aligned} \tag{4.24}$$

the closed-loop state $x(t)$ remains bounded in $\Omega_{\hat{\rho}}$ and is ultimately bounded in $\Omega_{\rho_{\min}}$ for $t \geq T$, under the proposed encrypted decentralized LMPC of Eq. (4.13)*.*

Proof. Firstly, within this proof, we establish the recursive feasibility of the optimization problem within each decentralized LMPC. Subsequently, under the optimized control actions of the encrypted decentralized LMPC of Eq. (4.13), we will prove the boundedness and convergence of the closed-loop state of the nonlinear system within the set $\Omega_{\hat{\rho}}$, extending the results of Theorem 4.1. Initially, we consider $x(t) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_{\min}}$. The input trajectories $\hat{\Phi}_j(\hat{x}(t_k)),\ j = 1, \ldots, N_{sys}$ for $t \in [t_k, t_{k+1})$ are feasible solutions to the optimization problem outlined in Eq. (4.13), as the input constraint of Eq. (4.13c) and the Lyapunov constraint of Eq. (4.13e) are both satisfied. Then, we consider $x(t) \in \Omega_{\rho_{\min}}$. The input trajectories $\hat{\Phi}_j(\tilde{x}(t_{k+i})),\ i = 0, 1, \ldots, N-1,\ j = 1, \ldots, N_{sys}$ for $t \in [t_k, t_{k+N})$ satisfy the constraints on the inputs in Eq. (4.13c) and the Lyapunov-based constraint of Eq. (4.13f). It has been proven in Theorem 4.1 that the states predicted by the LMPC process model of Eq. (4.13b) can remain within $\Omega_{\rho_{\min}}$ under the encrypted stabilizing controllers $\hat{\Phi}_j(\tilde{x})$ for time $t \geq T$. Thus, the optimization problem of each decentralized LMPC would be feasible for all $x_0 \in \Omega_{\hat{\rho}}$ and can be solved by recursive feasibility for $t \in [t_k, t_{k+1})$, i.e.,

$$\begin{aligned} &\frac{\partial V(x(t))}{\partial x_j} f_j(x(t), x(t) + \xi_1(t), \hat{u}_{d_j}(t_k)) \\ &\leq \frac{\partial V(x(t))}{\partial x_j} f_j(x(t), x(t) + \xi_1(t), \hat{\Phi}_j(\hat{x}(t_k))),\ \forall j = 1, \ldots, N_{sys} \end{aligned} \tag{4.25}$$

The control Lyapunov function for the overall system $V(x)$ may take the form of a linear combination of control Lyapunov functions for individual subsystems. In this representation, $V(x)$ is expressed as the sum of $V_j(x_j)$ for each subsystem, where V_j is assumed to be a function of x_j only. The time-derivative of the control Lyapunov function of the encrypted decentralized LMPC can be expressed as follows:

$$\dot{V} = \sum_{j=1}^{N_{sys}} \frac{\partial V(x(t))}{\partial x_j} f_j(x(t), x(t) + \xi_1(t), \hat{u}_{d_j}(t_k)) \tag{4.26}$$

Based on the Lyapunov constraint, the following inequality holds:

$$\begin{aligned} \dot{V} = &\sum_{j=1}^{N_{sys}} \frac{\partial V(x(t))}{\partial x_j} f_j(x(t), x(t) + \xi_1(t), \hat{u}_{d_j}(t_k)) \leq \\ &\sum_{j=1}^{N_{sys}} \frac{\partial V(x(t))}{\partial x_j} f_j(x(t), x(t) + \xi_1(t), \hat{\Phi}_j(\hat{x}(t_k))) \end{aligned} \tag{4.27}$$

From Eq. (4.26) and Eq. (4.27), the time-derivative of the control Lyapunov function under the encrypted decentralized LMPC satisfies the inequality,

$$\frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \hat{u}_d(t_k)) \leq \frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \hat{\Phi}(\hat{x}(t_k))) \tag{4.28}$$

However, from the results of Theorem 4.1 (Eq. (4.23)), it follows that the right-hand side of Eq. (4.28) is bounded as follows:

$$\frac{\partial V(x(t))}{\partial x} f(x(t), x(t) + \xi_1(t), \hat{u}_d(t_k)) \leq L'_x M_F \Delta + L'_\xi d_1 M_{d_1} + L'_q |e_t| - \frac{c_3}{c_2} \rho_s \leq -\epsilon_w \tag{4.29}$$

Thus, for the chosen time T, there exist positive real numbers Δ, d_1, d, and ϵ_w, such that the

following inequality holds,

$$L'_x M_F \Delta + L'_\xi d_1 M_{d_1} + L'_q |e_t| - \frac{c_3}{c_2}\rho_s \leq -\epsilon_w \; \forall \, t \in [0, T]$$

which implies that $\dot{V} \leq -\epsilon_w$ for any $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_s}$ for all $t_k \in [0, T]$. This establishes that, if the conditions of Eq. (4.24) are met, the closed-loop system state is always bounded in $\Omega_{\hat{\rho}}$, and it converges to $\Omega_{\rho_s} \subseteq \Omega_{\rho_{\min}}$ within time T, and remains there. This completes the proof for the stability of the system under the encrypted decentralized LMPC. □

Remark 4.4. *As discussed in Section* 4.3.3*, we employ the predictor feedback methodology outlined in Section* 4.3.4 *to achieve system stabilization in the presence of input delays. The stability analysis does not consider the perturbation caused by input delays. However, a similar approach to the one used to establish bounds on state delays, as demonstrated in Eq.* (4.21)*, could be employed to account for the influence of input delays. Incorporating input delay perturbations into the proof would establish a very stringent upper limit on the allowable value of* d_2*, rendering the proof valid only for relatively small input delays. As outlined in Eq.* (4.3)*, the perturbation resulting from input delays can be expressed as* $\xi_2(t) = u(t - d_2) - u(t)$*. Consequently, it becomes evident that, as* d_2 *approaches zero,* $\xi_2(t)$ *tends to zero as well. In the interest of maintaining a more generalized analysis with established bounds applicable even to substantial input delays, we have chosen to omit this consideration from the proof. Instead, we opt to address input delay challenges by employing a predictor, ensuring the validity of our analysis across a broader range of scenarios.*

4.3.4 Predictor feedback decentralized LMPC methodology

This subsection formulates a predictor feedback-based decentralized LMPC for the nonlinear system described in Eq. (4.1). A first-principles-based state predictor is integrated in the closed-loop system to compensate for the effect of input delays. At time t_k, where k is the sampling instance, the predictor of the j^{th} subsystem receives the quantized states $\hat{x}(t_k)$. It uses the control input trajectory $u_{d_j}(t)$ computed previously by the j^{th} LMPC, and estimates the control inputs for the other subsystems using the stabilizing control law, $\Phi(x)$, over t_k to $t_k + d_2$, to predict the state values of the entire system at $t_k + d_2$. Additionally, the LMPCs employ a DDE-based nonlinear process model specific to their subsystem. Thus, the predictor also transmits values of the states from time $t_k + d_2 - d_1$ to $t_k + d_2$, which are used by the DDE model to account for the state delays in the system. Within a decentralized control framework, where inter-controller communication is absent, the predictor of the j^{th} subsystem only has access to the control inputs computed by the j^{th} LMPC. Thus, an estimate of the control inputs of the other subsystems can be made through the stabilizing control law, utilizing state feedback. The inputs are assumed to be at their steady state values from time 0 to d_2. The j^{th} LMPC is then initialized with the shifted timescale $\bar{t}_k = t_k + d_2$ to calculate the optimal control input trajectory, u_{d_j}, from $\bar{t}_k$ to $\bar{t}_{k+N}$. The LMPC formulation with

the shifted time scale is described as follows:

$$\mathcal{J}_j = \min_{u_{d_j} \in S(\Delta)} \int_{\bar{t}_k}^{\bar{t}_{k+N}} L_j(\tilde{x}_j(t), u_{d_j}(t))\, \mathrm{d}t \tag{4.30a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}_j(t) = F_j(\tilde{x}(t), u_{d_j}(t)) \tag{4.30b}$$

$$u_{d_j}(t) \in U_j,\ \forall\, t \in [\bar{t}_k, \bar{t}_{k+N}) \tag{4.30c}$$

$$\tilde{x}(\bar{t}_k) = \hat{x}(\bar{t}_k) \tag{4.30d}$$

$$\dot{V}(\hat{x}(\bar{t}_k), u_{d_j}(\bar{t}_k)) \leq \dot{V}(\hat{x}(\bar{t}_k), \Phi_j(\hat{x}(\bar{t}_k))),$$

$$\text{if } \hat{x}(\bar{t}_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{4.30e}$$

$$V(\tilde{x}(t)) \leq \rho_{\min},\ \forall\, t \in [\bar{t}_k, \bar{t}_{k+N}),$$

$$\text{if } \hat{x}(\bar{t}_k) \in \Omega_{\rho_{\min}} \tag{4.30f}$$

Remark 4.5. *As mentioned earlier in Section* 4.3.3*, we employ the predictor feedback methodology outlined in Section* 4.3.4 *to achieve system stabilization in the presence of input delays. In the absence of a predictor, nominal to modest input delays can lead to an oscillatory convergence of the closed-loop system states around their respective steady states within* Ω_ρ *but outside* $\Omega_{\rho_{\min}}$*, while larger input delays can cause the state to exit* Ω_ρ*. However, with a predictor feedback methodology, the closed-loop states can be stabilized within* $\Omega_{\rho_{\min}}$ *even under large input delays. This is demonstrated in the example described in Section* 4.4*.*

4.4 Application to a nonlinear chemical process network operating at an unstable steady state

This section demonstrates the proposed encrypted decentralized control architecture on a nonlinear chemical process network with input and state delays, operating at an unstable steady state. A nonlinear dynamical model based on first-principles modeling fundamentals is developed for the state predictor and the LMPCs. This model is partitioned into N_{sys} subsystems to construct first-principles-based process models of the decentralized LMPC of each subsystem. Guidelines are established to implement the encrypted decentralized LMPC system in any nonlinear process with delays. We then conduct closed-loop simulations, employing the decentralized LMPC with and without the predictor feedback, and analyze the results.

4.4.1 Process description and model development

The process considered is the synthesis of ethylbenzene (EB) by reacting ethylene (E) and benzene (B) within two non-isothermal, well-mixed continuous stirred tank reactors (CSTRs) as depicted in Figure 4.2. The primary reaction, termed as "primary", is characterized as a second-order, exothermic, and irreversible reaction, in conjunction with two supplementary side reactions. The chemical reactions taking place are articulated as follows:

$$C_2H_4 + C_6H_6 \rightarrow C_8H_{10} \qquad \text{(primary)} \tag{4.31a}$$

$$C_2H_4 + C_8H_{10} \rightarrow C_{10}H_{14} \tag{4.31b}$$

$$C_6H_6 + C_{10}H_{14} \rightarrow 2C_8H_{10} \tag{4.31c}$$

Details of the steady-state values and model parameter values can be obtained from [39]. The dynamic model of the initial CSTR is described by the following mass and energy balance equations:

$$\dot{C}_{E_1}(t) = \frac{F_1 C_{Eo_1}(t-d_2) - F_{out_1} C_{E_1}(t)}{V_1} - r_{1,1} - r_{1,2} \tag{4.32a}$$

$$\dot{C}_{B_1}(t) = \frac{F_1 C_{Bo_1}(t-d_2) - F_{out_1} C_{B_1}(t)}{V_1} - r_{1,1} - r_{1,3} \tag{4.32b}$$

$$\dot{C}_{EB_1}(t) = \frac{-F_{out_1} C_{EB_1}(t)}{V_1} + r_{1,1} - r_{1,2} + 2r_{1,3} \tag{4.32c}$$

$$\dot{C}_{DEB_1}(t) = \frac{-F_{out_1} C_{DEB_1}(t)}{V_1} + r_{1,2} - r_{1,3} \tag{4.32d}$$

$$\dot{T}_1(t) = \frac{T_{1_o} F_1 - T_1(t) F_{out_1}}{V_1} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_1 C_p} r_{1,j} + \frac{Q_1(t-d_2)}{\rho_1 C_p V_1} \tag{4.32e}$$

The dynamic model of the second CSTR is represented by the following equations:

$$\dot{C}_{E_2}(t) = \frac{F_2 C_{Eo_2}(t-d_2) + F_{out_1} C_{E_1}(t-d_1)}{V_2} - \frac{F_{out_2} C_{E_2}(t)}{V_2} - r_{2,1} - r_{2,2} \tag{4.33a}$$

$$\dot{C}_{B_2}(t) = \frac{F_2 C_{Bo_2}(t-d_2) + F_{out_1} C_{B_1}(t-d_1)}{V_2} - \frac{F_{out_2} C_{B_2}(t)}{V_2} - r_{2,1} - r_{2,3} \tag{4.33b}$$

$$\dot{C}_{EB_2}(t) = \frac{F_{out_1} C_{EB_1}(t-d_1) - F_{out_2} C_{EB_2}(t)}{V_2} + r_{2,1} - r_{2,2} + 2r_{2,3} \tag{4.33c}$$

$$\dot{C}_{DEB_2}(t) = \frac{F_{out_1} C_{DEB_1}(t-d_1) - F_{out_2} C_{DEB_2}(t)}{V_2} + r_{2,2} - r_{2,3} \tag{4.33d}$$

$$\dot{T}_2(t) = \frac{T_{2_o} F_2 + T_1(t-d_1) F_{out_1} - T_2(t) F_{out_2}}{V_2} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_2 C_p} r_{2,j} + \frac{Q_2(t-d_2)}{\rho_2 C_p V_2} \tag{4.33e}$$

where the reaction rates are calculated by the following expressions:

$$r_{i,1} = k_1 \mathrm{e}^{\frac{-E_1}{RT_i(t)}} C_{E_i}(t) C_{B_i}(t) \tag{4.34a}$$

$$r_{i,2} = k_2 \mathrm{e}^{\frac{-E_2}{RT_i(t)}} C_{E_i}(t) C_{EB_i}(t) \tag{4.34b}$$

$$r_{i,3} = k_3 \mathrm{e}^{\frac{-E_3}{RT_i(t)}} C_{DEB_i}(t) C_{B_i}(t) \tag{4.34c}$$

and $i = \{1, 2\}$ is the reactor index. The state variables are the concentration of ethylene, benzene, ethylbenzene, di-ethylbenzene, and the reactor temperature for each CSTR in deviation terms, that is: $x^\top = [C_{E_1} - C_{E_{1s}},\ C_{B_1} - C_{B_{1s}},\ C_{EB_1} - C_{EB_{1s}},\ C_{DEB_1} - C_{DEB_{1s}},\ T_1 - T_{1s},\ C_{E_2} - C_{E_{2s}},\ C_{B_2} - C_{B_{2s}},\ C_{EB_2} - C_{EB_{2s}},\ C_{DEB_2} - C_{DEB_{2s}},\ T_2 - T_{2s}]$. The subscript "$s$" denotes the steady-state value. The state delay, representing the time needed to transport the output of the initial CSTR to the second CSTR, is set at $d_1 = 0.5\,\mathrm{min}$. The rate of heat removal for the two reactors $[Q_1 - Q_{1s},\ Q_2 - Q_{2s}]$ and inlet feed concentrations for each reactor, $[C_{E_{o1}} - C_{E_{o1s}}, C_{B_{o1}} - C_{B_{o1s}}, C_{E_{o2}} - C_{E_{o2s}}, C_{B_{o2}} - C_{B_{o2s}}]$, are the manipulated inputs with input delay $d_2 = 1\,\mathrm{min}$. These inputs are bounded by the closed sets, $[-10^4, 2 \times 10^3]$ kW, $[-1.5 \times 10^4, 5 \times 10^3]$ kW, $[-2.5, 2.5]$ kmol/m^3, $[-2.5, 2.5]$ kmol/m^3, $[-3, 3]$ kmol/m^3, and $[-3, 3]$ kmol/m^3, respectively. To determine the stability of the chosen steady-state, an open loop simulation was performed where the control inputs were maintained at their steady state values, and the system states were initialized at a point close to their operating steady-state within $\Omega_{\rho_{\min}}$. After a finite duration of process time, the states exited the stability region, Ω_ρ, and converged to another steady state, implying that the chosen steady-state is an unstable one. Furthermore, the rationale for choosing this steady-state was its ability to provide a high steady-state concentration ($4.22\,\mathrm{kmol/m^3}$) of the desired

product, ethyl benzene, at reasonable operating conditions, at the outlet of reactor 2, making it the economically optimal steady state to operate at.

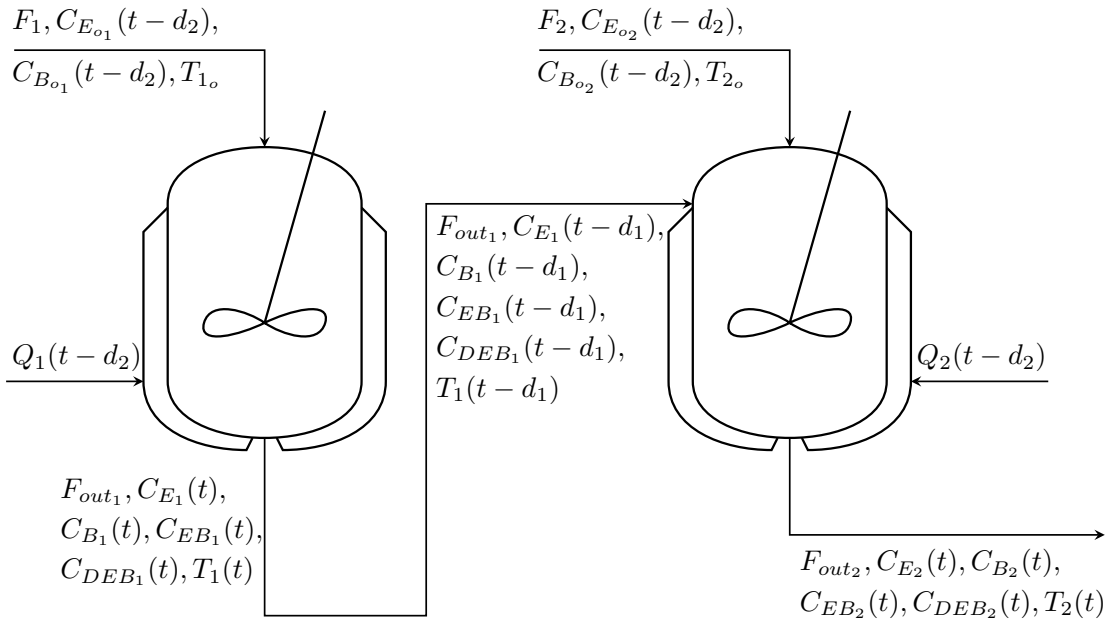

Figure 4.2: Process schematic featuring two CSTRs connected in series.

We create two decentralized LMPCs in our design. The first LMPC (LMPC 1) utilizes the first-principles-based model specific to subsystem 1, which corresponds to the dynamic model of CSTR 1 (Eq. (4.32)), while the second LMPC (LMPC 2) employs a first-principles-based model specific to subsystem 2, which corresponds to the dynamic model of CSTR 2 (Eq. (4.33)). LMPC 1 does not require complete state feedback, given that the dynamics of its subsystem are entirely independent of subsystem 2. However, the evolution of the states within the second CSTR is influenced by the states of the first CSTR. Thus, LMPC 1 receives $x_1 = [C_{E_1} - C_{E_{1s}},\ C_{B_1} - C_{B_{1s}},\ C_{EB_1} - C_{EB_{1s}},\ C_{DEB_1} - C_{DEB_{1s}},\ T_1 - T_{1s}]^\top$ and optimizes the control inputs $u_1 = [C_{E_{o1}} - C_{E_{o1s}}, C_{B_{o1}} - C_{B_{o1s}}, Q_1 - Q_{1s}]^\top$. LMPC 2 receives full state feedback x, and optimizes the control inputs $u_2 = [C_{E_{o2}} - C_{E_{o2s}}, C_{B_{o2}} - C_{B_{o2s}}, Q_2 - Q_{2s}]^\top$. The control objective is to operate both

CSTRs at their unstable equilibrium point through the encrypted decentralized control scheme, employing quantized states and inputs for computation and actuation.

4.4.2 Encrypting the decentralized control architecture

Before implementing encryption–decryption into a process, the selection of parameters, namely d, l_1, and l_2 is performed. Based on the extreme feasible states and inputs, the integer bit count $l_1 - d$ is derived. The upper limit in the $\mathbb{Q}_{l_1,d}$ set is obtained via the formula $2^{l_1-d-1} - 2^{-d}$, whereas the lower limit is -2^{l_1-d-1}. The choice of the quantization parameter d, representing the fractional bit count, rests on the desired accuracy and range of state and input values. Additionally, l_2 is chosen to exceed l_1. Accordingly, for the example in this section, $l_1 - d$ is calculated to be 16, from which l_1 and d are then fixed. Within the set $\mathbb{Q}_{l_1,d}$, rational numbers are separated by a resolution of 2^{-d}. For simulation purposes, we use, $d = 8$. For $d = 8$, $l_1 = 24$ and we select $l_2 = 30$. The Paillier Encryption procedure is implemented through Python's "phe" module, PythonPaillier [21]. For solving the constrained non-convex optimization problem in the LMPCs within the decentralized control structure, we leverage the Python module of the IPOPT software [83].

While deciding the sampling time (Δ) for an encrypted decentralized system, it is crucial to ensure that it exceeds the total time required for encryption–decryption of the states and control inputs, time required by the predictor to predict the states after the input delay, and the time needed to compute the control inputs at each sampling instance for the considered quantization parameter d, for any subsystem, as these computations would occur concurrently in different edge computing

units. Mathematically,

$$\Delta > \max\left(\text{Encryption-decryption time}\right)_j + \max\left(\text{Control input computation time}\right)_j + \max\left(\text{State-prediction time}\right)_j \tag{4.35}$$

where $j = \{1, \ldots, N_{sys}\}$ represents the control subsystem. Considering the above criteria, the sampling time Δ is chosen as 30 seconds in the discussed example.

To calculate the cost function of the LMPCs over the prediction horizon, the integration step $h_c = 10^{-2} \times \Delta$ is chosen. The positive definite matrix P in the control Lyapunov function $V = x^\top P x$ is selected as $\text{diag}\ [250\ \ 500\ \ 500\ \ 1000\ \ 2.5\ \ 250\ \ 250\ \ 500\ \ 1000\ \ 2.5]$, from extensive simulations. The LMPCs employ a prediction horizon of $N = 3$ sampling periods. The stability criterion is defined as $\rho = 1000$, while $\rho_{\min} = 2$ is the smaller level set of the Lyapunov function where the state is desired to be confined. The weight matrices in the cost function of LMPCs are chosen as $Q_1 = \text{diag}\ [2000\ \ 2000\ \ 5000\ \ 5\ \ 50]$, $Q_2 = \text{diag}\ [1000\ \ 1000\ \ 2500\ \ 5\ \ 135]$, $R_1 = \text{diag}\ [1\ \ 1\ \ 5 \times 10^{-6}]$, and $R_2 = \text{diag}\ [20\ \ 15\ \ 2.5 \times 10^{-4}]$. The cost function is defined as $L_j(x_j, u_j) = x_j^\top Q_j x_j + u_j^\top R_j u_j$ where $j = 1, 2$ represents the LMPC j. As di-ethylbenzene, the undesired product, is present in trace amounts in both CSTRs, its trajectories are not depicted.

4.4.3 Simulation results of the encrypted decentralized control architecture

The proposed encrypted decentralized control architecture is applied to a nonlinear chemical process with state and input delays. Figures 4.3 to 4.5 and Figures 4.6 to 4.8 depict the results for the encrypted decentralized LMPC system without and with predictor feedback, respectively.

In the absence of a predictor, the states and inputs of both CSTRs show considerable oscil-

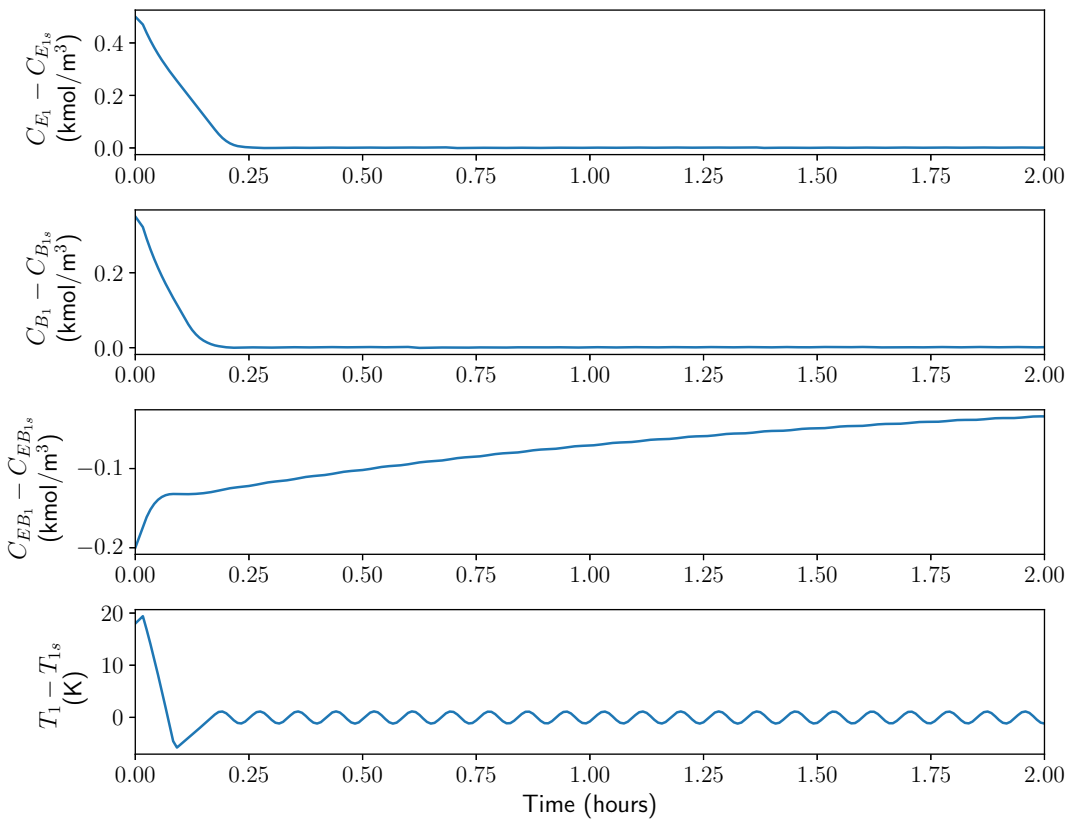

Figure 4.3: State profiles of CSTR 1 under the encrypted decentralized LMPC for state delay $d_1 = 0.5\,\text{min}$, and input delay $d_2 = 1\,\text{min}$.

lations, as shown in Figures 4.3 to 4.5. Additionally, the temperatures of both CSTRs overshoot their set-points. With the addition of the state predictor, the oscillations in both states and inputs are negligible as observed in Figures 4.6 to 4.8. Furthermore, there is no overshoot of the temperature in CSTR 1, and the overshoot in temperature is decreased for CSTR 2. Moreover, the inclusion of the predictor enables us to achieve convergence of the states within the targeted stability region, denoted as $\Omega_{\rho_{\min}}$. This was not attainable solely with the encrypted decentralized LMPC. While the latter stabilizes the states within Ω_ρ, it falls short of achieving convergence within the desired stability region after two hours of process time.

To measure the computational time for computing the control inputs in the decentralized

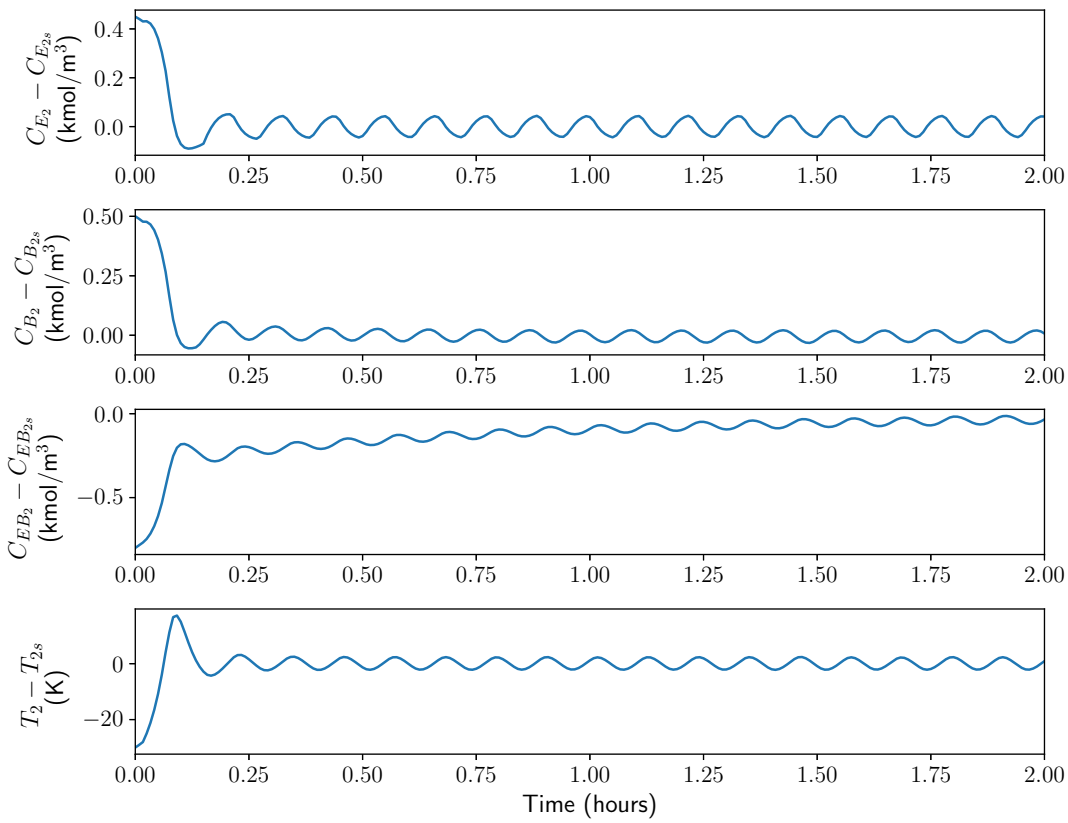

Figure 4.4: State profiles of CSTR 2 under the encrypted decentralized LMPC for state delay $d_1 = 0.5\,\text{min}$, and input delay $d_2 = 1\,\text{min}$.

MPC, we recorded the maximum time taken by the 2 MPCs at each sampling instance. On average, the decentralized controllers spent 2.49 seconds on control input computation at every sampling instance, whereas the centralized controller averaged 10.75 seconds. We ensured that the control input computation time remained below the 30-second sampling interval for all sampling times. These results demonstrate the computational efficiency of a decentralized MPC over a centralized MPC. Furthermore, the normalized sum of the control cost function for the centralized and decentralized MPCs without delays was recorded as 1 and 0.9798, respectively. The reason for a slightly better performance under the decentralized MPC can be attributed to the fact that the process network has a sequential flow sheet with 2 CSTRs in series, which makes the decentralized MPC a

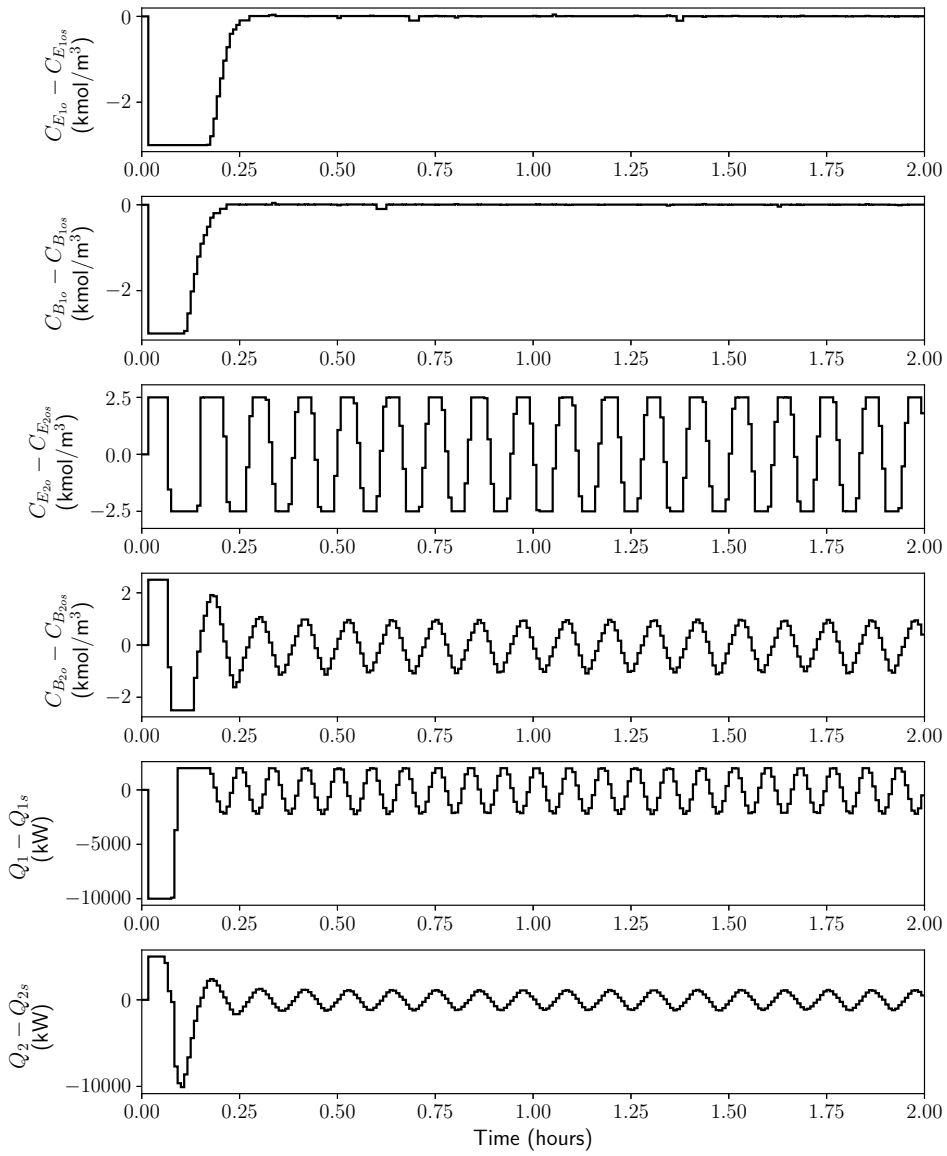

Figure 4.5: Control input profiles under the encrypted decentralized LMPC for state delay $d_1 = 0.5\,\text{min}$, and input delay $d_2 = 1\,\text{min}$.

more suitable, well-conditioned choice than the centralized MPC with respect to the optimization problem solution. These results validate the effectiveness of the proposed decentralized LMPC framework in comparison to a centralized LMPC for this particular process network.

Remark 4.6. *During the initial delay period, where control input information is not yet available, we assume steady-state values for the control inputs. This assumption results in a sharp increase*

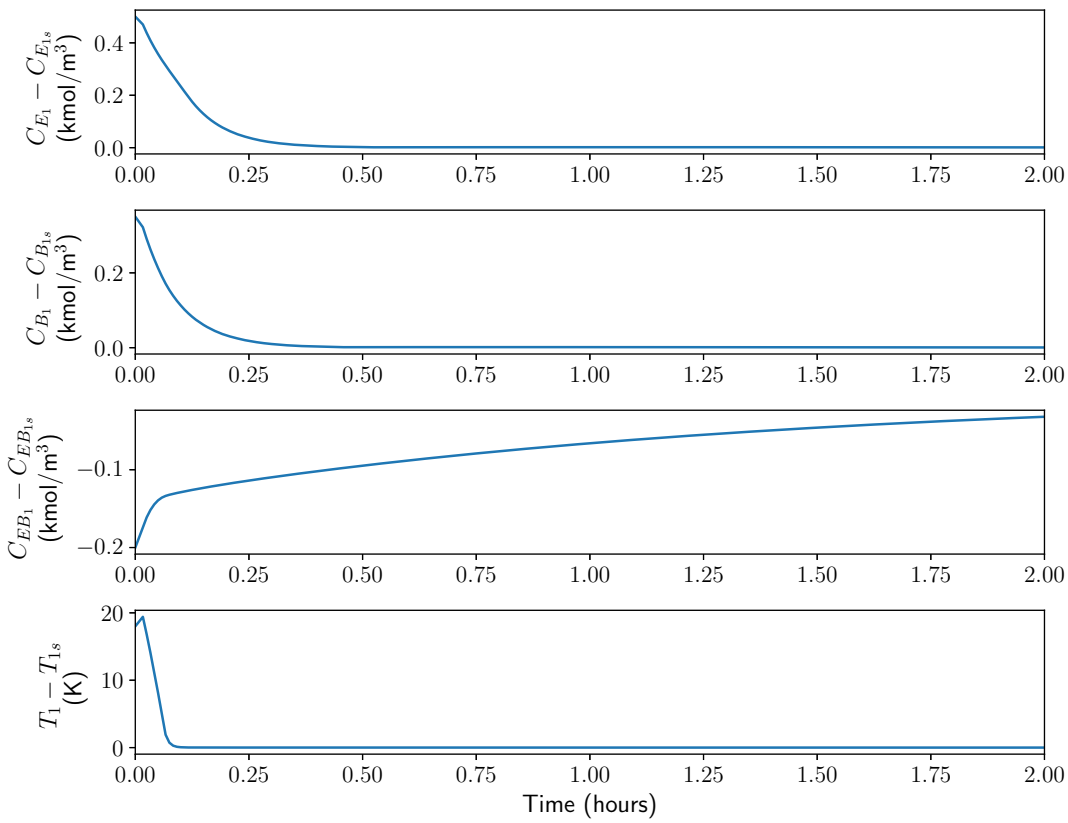

Figure 4.6: State profiles of CSTR 1 under the encrypted decentralized LMPC with predictor feedback for state delay $d_1 = 0.5\,\text{min}$, and input delay $d_2 = 1\,\text{min}$.

that would not typically occur during continuous operation. To mitigate such abrupt changes in control inputs, one approach is to introduce a constraint on the maximum allowable change in applied control inputs between sampling instances. This constraint can help smooth the transition between steady-state values and actual control inputs initially, and also reduce sudden spikes or fluctuations in the system's behavior for the remainder of the operation.

Remark 4.7. *The encrypted decentralized LMPC explored in this study involved encrypting and decrypting data as outlined in Figure* 4.1*, which can lead to errors due to quantization. [81] demonstrated quantization effects in the context of a first-principles-based MPC. Additionally, [39] highlighted the potential for quantization-induced errors to exceed model mismatch errors*

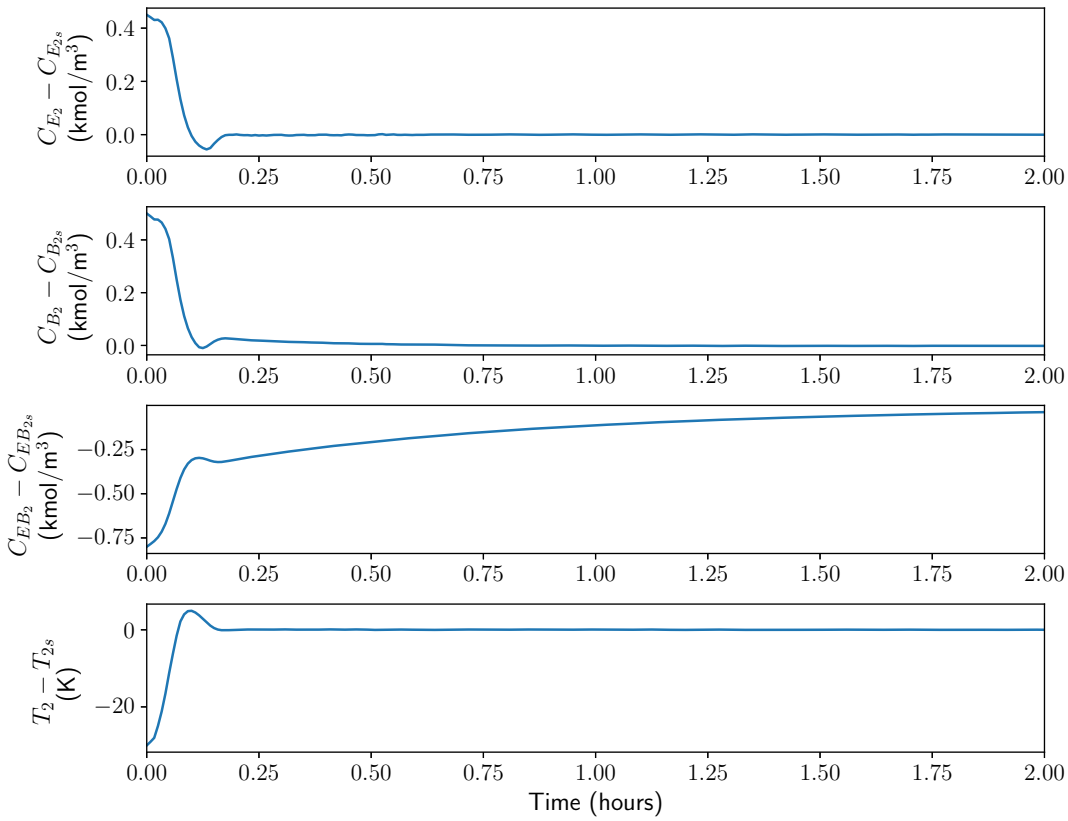

Figure 4.7: State profiles of CSTR 2 under the encrypted decentralized LMPC with predictor feedback for state delay $d_1 = 0.5\,\text{min}$, and input delay $d_2 = 1\,\text{min}$.

when different models are employed in the MPC and in the controlled process. To minimize the quantization error, both works recommended using a higher quantization parameter d*. With* $d = 8$*, both works reported almost identical closed-loop results with encryption compared to without encryption. Thus, we have used the quantization parameter,* $d = 8$ *for all simulations in this work.*

Remark 4.8. *In this work, we have assumed the same value of the input delay for all the control inputs applied to the nonlinear process. However, if the input delay values are different for certain control inputs, in the proposed encrypted decentralized control structure, the subsystems can be partitioned in a manner such that the control inputs manipulated by each subsystem have the same input delay values. Thus, the predictor of a particular subsystem would predict the states up to the*

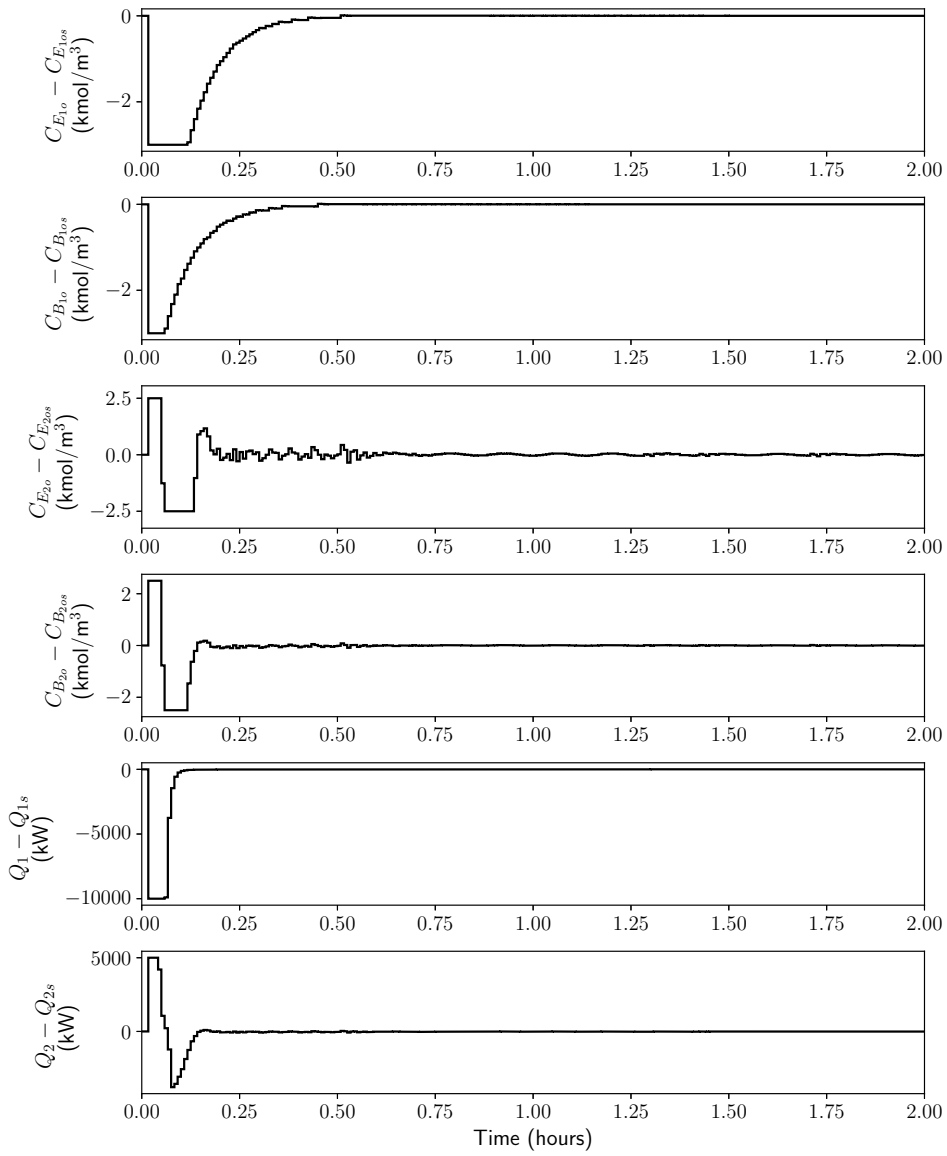

Figure 4.8: Control input profiles under the encrypted decentralized LMPC with predictor feedback for state delay $d_1 = 0.5\,\text{min}$, and input delay $d_2 = 1\,\text{min}$.

corresponding input delay value of the control inputs manipulated by that subsystem.

Remark 4.9. *Although the LMPC and predictor models used in this work are first-principles-based, data-based models employing artificial neural networks can also be used in the predictor and LMPC. [5] used machine-learning-based models for the predictor and LMPC while simulating a first-principles-based process with state and input delays, showcasing the effectiveness of the*

predictor in the presence of plant/model mismatch.

4.5 Conclusion

In this chapter, we devised and applied an encrypted decentralized control architecture to a large-scale nonlinear chemical process network with input and state delays. A stability analysis of the encrypted decentralized MPC applied to a nonlinear system with state delays was conducted, yielding bounds on the errors due to quantization, state delays, and sample-and-hold implementation of the controller. Based on these bounds, the system can be stabilized within the desired stability region. We established guidelines to implement this control structure in any nonlinear process, such as selection of parameters l_1, l_2, and d for quantization, and the sampling time criterion. The encrypted decentralized LMPC employs a DDE model to account for state delays in the process. Closed-loop simulations are compared with and without the incorporation of a predictor into the LMPC design, where the predictor predicts the state values after the input delay period. A significant improvement in the closed-loop performance was observed with the integration of the predictor, as the states and inputs converged to their steady state values with negligible oscillations. Also, with the inclusion of the predictor, states converged within the desired stability region represented by the level set $\Omega_{\rho_{\min}}$. However, without the predictor, the states only stabilize within the larger level set Ω_ρ and with oscillations. Thus, by employing the encrypted decentralized LMPC with predictor feedback, we were able to reduce the computation time and complexity of the control problem, improve the closed-loop performance, and enhance the cybersecurity of the control system.

Chapter 5.1

Encrypted distributed model predictive control with state estimation for nonlinear processes

5.1.1 Introduction

Industrial control systems for large-scale processes have been subject to extensive research over the past decades, with the primary objectives of enhancing operational safety, promoting environmental sustainability, optimizing profitability, and economizing on utility costs. Nonetheless, the evolution of technology has led to the integration and interlinking of industrial control systems with corporate networks and the internet, to create cyber-physical systems that have streamlined monitoring, control, and automation of complex processes, enhancing productivity and operational efficiency. However, the increased connectivity and linking of these systems have made them vulnerable to cyberthreats, given their extensive reliance on networked communication. A breach or

compromise in these systems can have severe consequences, including the disruption of essential services, physical damage, financial losses, and are even a threat to public safety. As a result, the past few years have witnessed a surge in research efforts directed towards enhancing the cybersecurity of industrial control systems.

Recent developments in cyberattack techniques underscore the need to establish robust cybersecurity [31]. Dealing with cybersecurity issues within industrial control systems is mainly within the realm of operational technology (OT). While there have been notable advancements in improving cybersecurity in the information technology (IT) sector, which centers on the software elements of systems, including aspects like network architecture and data management, cybersecurity within the OT domain is currently trailing behind [16]. Numerous real-world examples highlight the need of cybersecurity in networked cyber–physical systems and SCADA (Supervisory Control and Data Acquisition) systems. These include the 2015 cyberattacks on SCADA controls responsible for managing the power grid in Ukraine, leading to widespread power outages [45]. Likewise, in the 2021 DarkSide ransomware attack on Colonial Pipeline, cyberattackers encrypted its networked communication and demanded a ransom for the decryption keys. As a result, Colonial Pipeline was compelled to suspend its operations, resulting in interruptions to fuel distribution and financial losses [82].

Traditional control systems, like proportional-integral-derivative (PID) control, have long been used in chemical plants to control processes with a decentralized structure. In this setup, each controller uses one process measurement and calculates actions to control that specific state at its desired set point. However, PIDs do not consider how the controlled variable interacts with other states, limiting their ability to optimize control inputs for a multi-input-multi-output (MIMO)

system. To address this limitation, model predictive controllers (MPCs) have been applied to manage complex processes. MPCs employ models, derived from first principles, data, or mathematical representations, to predict future states within a specified horizon. They then optimize control inputs using real-time sensor feedback while accounting for interactions among all the process states and inputs. This approach not only elevates control precision but also mitigates utility expenses, ultimately enhancing overall process performance.

However, acquiring sensor measurements for all states in large processes can incur significant costs. Furthermore, installing the necessary instrumentation and equipment to collect and transmit measurements may not always be feasible, particularly in specific areas of the plant or process. As a result, extensive research has been carried out on state estimation techniques, enabling real-time prediction of unmeasured states though deterministic and stochastic estimation methods. Notably, the extended Kalman Filter (EKF) and extended Luenberger observer (ELO) stand out as commonly used state estimators for nonlinear processes. The EKF utilizes a stochastic approach, employing a linearized approximation from continuous time to a discrete-time system to estimate the state. This method can account for system and measurement noise using probabilistic approaches. Conversely, the ELO adopts a deterministic approach, using nonlinear model dynamics to estimate states without explicitly addressing stochastic disturbances or measurement noise. While the EKF can account for sensor noise better, the ELO handles nonlinearity by directly incorporating it in the observer equations. More details on the advantages, drawbacks, and similarities about various state estimation methods has been discussed in the works of [69] and [4]. To attain the desired performance using these methods, a mathematical model for the specific system is typically required to describe process dynamics within a defined operating range. Nevertheless,

when integrated with MPC, the MPC model can be extended for use by the state estimator, and vice versa, enabling a collaborative and effective solution.

Since MPCs employ nonlinear optimization for control input optimization in nonlinear processes, in large-scale systems, where numerous control inputs must be calculated, the control problem can become too extensive and intricate to be solved within the given sampling time. As a response, decentralized and distributed MPC strategies have been introduced to break down the complex problem into smaller segments, handled by different computing units. In such arrangements, the system to be controlled is partitioned into smaller subsystems, where the control input of each subsystem is computed separately. Decentralized MPCs compute control inputs for their respective subsystems without any knowledge of the control inputs being applied by other subsystems. This limits the controller from taking into account interactions among different process subsystems and only considers interactions within its specific subsystem. In contrast, distributed controllers share information about the control inputs computed for their subsystem, enabling other controllers to optimize their control inputs accordingly. This collaborative approach improves the handling of interdependencies among various process subsystems.

Significant research efforts have been devoted to various domains of cybersecurity, and process control, including the development of machine learning-based cyberattack detectors [2, 26], the implementation of nonlinear encrypted centralized MPCs [81], the utilization of sequential and iterative DMPCs [52], and the application of nonlinear state estimators [43, 89]. However, to the best of our knowledge, the development of distributed control systems that employ encrypted networked communication for large-scale nonlinear processes with partial state feedback remains an unexplored area, prompting our proposal for a novel control structure to address this challenge.

Specifically, we propose a distributed control structure comprising a set of Lyapunov-based MPCs, integrated with an extended Luenberger observer, utilizing encrypted networked communication. In this configuration, we assume the presence of secure edge computers responsible for computing control inputs and receiving and transmitting encrypted signals. Integrating observer-based state estimation within this setup serves to provide each LMPC with complete state information in real-time. To address interactions within different subsystems in large processes and reduce the complexity associated with centralized control problems, we employ a distributed MPC. Further, the incorporation of encryption within the networked communication channels enhances cybersecurity as each edge computing unit receives and transmits encrypted wireless signals.

The remainder of this chapter is structured as follows: In Section 5.1.2, we provide an overview of various aspects, including notation, the considered class of nonlinear systems, system stabilizability assumptions, the formulation of the extended Luenberger observer, the employed encryption cryptosystem, and the implications of quantization. Section 5.1.3 delves into the design of the encrypted distributed MPC, outlining the formulation of sequential and iterative DMPCs utilizing state estimates from the observer, and further detailing the extended Luenberger observer. In Section 5.1.4, we present and discuss closed-loop simulations for a nonlinear chemical process network with partial state feedback in the presence of sensor noise with the encrypted sequential and iterative DMPCs. In Section 5.1.5, we conduct a comparative assessment of the encrypted control strategies, encompassing centralized, decentralized, and distributed MPCs.

5.1.2 Preliminaries

5.1.2.1 Notation

The symbol $\|\cdot\|$ represents the Euclidean norm of a vector. $x^\top$ denotes the transpose of a vector x. $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ represent the sets of real numbers, integers, and natural numbers, respectively. $\mathbb{Z}_M$ denotes the additive groups of integers modulo M. Set subtraction is indicated by the symbol "\", where $A \backslash B$ represents the set of elements that are in set A but not in set B. A function, $f(\cdot)$, falls under the class $\mathcal{C}^1$ if it is continuously differentiable within its defined domain. The term $\mathrm{lcm}(i, j)$ denotes the least common multiple of the integers i and j, while $\gcd(i, j)$ signifies the greatest common divisor, that divides i and j without any remainder.

5.1.2.2 Class of systems

This study is centered on multi-input multi-output (MIMO) systems, which are characterized by a category of continuous-time nonlinear systems represented in state-space form as follows:

$$\dot{x} = F(x, u) = f(x) + g(x)u \tag{5.1.1a}$$

$$y = h(x) + w \tag{5.1.1b}$$

The state vector is denoted by $x = [x_1, \ldots, x_n] \in \mathbb{R}^n$, while $u \in \mathbb{R}^m$ represents the control input vector bounded by the set, $U \subset \mathbb{R}^m$. The output vector consisting of the state measurements that are continuously sampled is $y = [y_1, \ldots, y_q] \in \mathbb{R}^q$, and $w \in \mathbb{R}^q$ is the measurement noise vector. $F(x, u)$ is a nonlinear function with respect to x and u, rendering the origin as a steady state of

Eq. (5.1.1). Without loss of generality, we assume the initial time as zero ($t_0 = 0$). The functions $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are matrices of dimension $n \times 1$, $n \times m$, and $q \times 1$, respectively. Additionally, we define the set $S(\Delta)$ as the set of piece-wise constant functions characterized by a period of Δ. We consider $j = 1, \ldots, N_{sys}$ sub-systems, where each subsystem j is regulated only by inputs u_j but potentially impacted by inputs of other subsystems due to coupling between subsystems. The control input vector for the j^{th} subsystem is $u_j \in \mathbb{R}^{m_j}$. $u = [u_1^\top \ldots u_{N_{sys}}^\top]^\top \in \mathbb{R}^m$ is the control input vector for the entire system, with $m = \sum_{j=1}^{N_{sys}} m_j$. The control input vector constraints are $u_j \in U_j := \{u_{\min,j_i} \leq u_{j_i} \leq u_{\max,j_i}, \forall i = 1, 2, \ldots, m_j\} \in \mathbb{R}^{m_j}, \forall j = 1, \ldots, N_{sys}$. Hence, the set U that constrains the control input vector for the entire system is formed by the union of sets U_j, where $j = 1, \ldots, N_{sys}$.

5.1.2.3 Extended Luenberger observer

The extended Luenberger observer (ELO) was introduced as a natural extension of the Luenberger observer, originally developed based on a linear approximation of processes [23, 89]. The primary objective of a state observer, such as the ELO, is to estimate the unmeasured internal states of a given system. This estimation is achieved by leveraging the available measured states from the process, in combination with the applied inputs. The formulation of the Extended Luenberger observer for a nonlinear system is expressed through Eq. (5.1.2), presenting a means to capture and estimate the system's unmeasured internal states in the following manner:

$$\dot{\bar{x}} = F(\bar{x}, u) + K(y - h(\bar{x})) \tag{5.1.2}$$

where $\bar{x} \in \mathbb{R}^n$ represents the estimated state vector, and the observer gain matrix is $K \in \mathbb{R}^{n \times q}$. Eq. (5.1.2) comprises two key components: the initial term corresponds to the process model dependent on the estimated states and applied control inputs, while the final term serves as the output prediction error, functioning as a correction term.

The objective of the ELO is to minimize the estimation error, $e = x - \bar{x}$, in which the time-derivative of the error is determined by the following equation [23]:

$$\dot{e} = F(\bar{x} + e, u) - F(\bar{x}, u) - K(h(\bar{x} + e) - h(\bar{x})) \tag{5.1.3}$$

For the estimation error, e, to decay to zero, the time-derivative of the error (shown in Eq. (5.1.3)) must also decay to zero. Therefore, the observer gain matrix K must be designed accordingly. To design K, Eq. (5.1.3) can be simplified to the following equation by linearizing the process model at a fixed point:

$$\dot{e} = (A - KL)e \tag{5.1.4}$$

where $A = \frac{\partial F(x,u)}{\partial x}\big|_{x=\bar{x}}$ and $L = \frac{\mathrm{d}h(x)}{\mathrm{d}x}\big|_{x=\bar{x}}$ are linearized terms of the nonlinear system evaluated at a specific reference point (in general, $L = \partial h(x,u)/\partial x|_{x=\bar{x}}$), typically the operating steady state of the system. Subsequently, the selection of the observer gain matrix K is conducted in a manner that ensures that all the eigenvalues of the matrix $A - KL$ have strictly negative real components.

5.1.2.4 Stability assumptions

Based on how the overall large-scale system is partitioned, there may exist interacting dynamics between the subsystems, as the states and control inputs of one subsystem may impact the states

and control inputs of other subsystems. Accounting for these interactions, we assume the existence of an observer and feedback stabilizing control law $u = \Phi(\bar{x})$ for the overall system with $u_j = \Phi_j(\bar{x}) \in U_j$, which regulate the individual subsystems $j = 1, \ldots, N_{sys}$, such that the origin of the overall system of Eq. (5.1.1) is rendered exponentially stable. This signifies the presence of a $\mathcal{C}^1$ control Lyapunov function $V(x)$ for which the following inequalities hold for all $x, \bar{x} \in \mathbb{R}^n$ within an open region D surrounding the origin:

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2, \tag{5.1.5a}$$

$$\frac{\partial V(x)}{\partial x} f(x, \Phi(\bar{x})) \leq -c_3|x|^2, \tag{5.1.5b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq c_4|x| \tag{5.1.5c}$$

where c_1, c_2, c_3, and c_4 are positive constants. $\Phi(\bar{x}) = [\Phi_1(\bar{x})^\top, \ldots, \Phi_{N_{sys}}(\bar{x})^\top]^\top$ is the vector concatenating the stabilizing feedback control laws for all N_{sys} subsystems. For the nonlinear system described by Eq. (5.1.1), the region of closed-loop stability can be defined as a level set, Ω_ρ, of the control Lyapunov function V, such that $\Omega_\rho := \{x \in D | V(x) \leq \rho\}$, where $\rho > 0$. Hence, originating from any initial condition within Ω_ρ, the control input, $\Phi(\bar{x})$, guarantees that the state trajectory of the closed-loop system remains within Ω_ρ.

Remark 5.1.1. *The assumption of an output feedback controller satisfying Eq.* (5.1.5) *involves two key requirements. First, it mandates that the observer states remain bounded within the region Ω_ρ. Second, it necessitates that the estimated error, denoted as e and defined as the difference between x and $\bar{x}$, converges to zero within a finite timeframe, regardless of the initial condition within Ω_ρ.*

To ensure the fulfillment of these prerequisites, a series of random closed-loop trajectories are generated for the nonlinear system described in Eq. (5.1.1) *under the observer and state feedback controller, and it is ensured that all trajectories converge within* Ω_ρ *in a finite number of sampling periods with* $e \to 0$*. More details regarding the observer and controller tuning are presented in Section* 5.1.4.

5.1.2.5 Paillier cryptosystem

In this research, we employ the Paillier cryptosystem [67] to encrypt signals, specifically state measurements (x) and control inputs (u), transmitted to and from the controllers. Although we do not make use of the semi-homomorphic property of additive homomorphism within the Paillier cryptosystem, we employ it so that traditional controllers, such as proportional-integral controllers, which can conduct computations in an encrypted space, can be integrated into the overall control architecture if required. The encryption procedure is initiated by generating the public and private key. The public key is used to encrypt integer messages into ciphertexts, and the private key is employed to decrypt ciphertexts and retrieve the original integer messages. The process of generating the public and private key can be outlined as follows:

1. Choose two large prime integers (p and q) randomly, ensuring $\gcd(pq, (p-1)(q-1)) = 1$.
2. Compute $M = pq$.
3. Choose an arbitrary integer $\bar{g}$ such that $\bar{g} \in \mathbb{Z}_{M^2}$, which is the multiplicative group of integers modulo M^2.
4. Compute $\lambda = \mathrm{lcm}(q-1, p-1)$.

5. Specify $\bar{L}(x) = (x-1)/M$.

6. Verify the existence of the subsequent modular multiplicative inverse,

   $u = (\bar{L}(\bar{g}^{\lambda} \bmod M^2))^{-1} \bmod M$.

7. If the inverse does not exist, revisit step 3 and select an alternate value of $\bar{g}$. If the inverse exists, $(M, \bar{g})$ is the public key and (λ, u) is the private key.

Once the keys are acquired, the public and private keys are distributed to authorized recipients for encryption and decryption, respectively. The encryption process is as follows:

$$E_M(m, r) = c = \bar{g}^m r^M \bmod M^2 \tag{5.1.6}$$

where r is a randomly selected integer from the set $\mathbb{Z}_M$, and c represents the ciphertext achieved through the encryption of m. The decryption procedure is as follows:

$$D_M(c) = m = \bar{L}(c^{\lambda} \bmod M^2)u \bmod M \tag{5.1.7}$$

Remark 5.1.2. *The significance of encryption lies in safeguarding data privacy against potential cyberattacks, particularly sophisticated attacks that might go undetected by traditional cybersecurity measures. In scenarios where constant values are transmitted during steady-state operations, conventional methods might result in the transmission of the same values after data transformations, mathematical operations or mapping of data to a certain set. However, in encryption, the generation of a random number each time data is encrypted ensures that identical numbers, when encrypted, yield distinct ciphertexts, bolstering cybersecurity measures significantly.*

Remark 5.1.3. *Various encryption methods, such as symmetric encryption, fully homomorphic encryption, and partially homomorphic encryption, can be employed to secure data. Symmetric encryption, like AES (Advanced Encryption Standard), is a non-homomorphic encryption technique that does not allow mathematical operations within an encrypted space. In contrast, fully homomorphic encryption, exemplified by schemes like BGV (Brakerski-Gentry-Vaikuntanathan), permits both addition and multiplication operations within an encrypted environment. Meanwhile, partially homomorphic encryption enables either multiplication or addition operations within the encrypted domain. For instance, the Paillier cryptosystem allows addition operations in an encrypted space. While our work does not utilize the semi-homomorphic property of the Paillier cryptosystem, it has been recently considered for integrating linear controllers like Proportional-Integral (PI) control in large-scale systems alongside nonlinear controllers such as MPCs. This integration allows for control input computations for the linear controller within an encrypted space without decryption, as demonstrated in the work of [41].*

5.1.2.6 Quantization

To use the Paillier cryptosystem, data to be encrypted must be in the form of natural numbers in $\mathbb{Z}_M$. However, the signal values before encryption are in floating-point. Consequently, we employ quantization, mapping the floating-point numbers into $\mathbb{Z}_M$ [19]. Using a signed fixed-point binary representation, we create a set, $\mathbb{Q}_{l_1,d}$, with parameters l_1 and d. These parameters define the total bit count (integer and fractional) and the fractional bits, respectively. The $\mathbb{Q}_{l_1,d}$ set encompasses rational numbers from -2^{l_1-d-1} to $2^{l_1-d-1} - 2^{-d}$, separated by 2^{-d}. A rational number q in $\mathbb{Q}_{l_1,d}$ can be expressed as $q \in \mathbb{Q}_{l_1,d}$, where $\exists \beta \in \{0,1\}^{l_1}$, and $q = -2^{l_1-d-1}\beta_{l_1} + \sum_{i=1}^{l_1-1} 2^{i-d-1}\beta_i$. To

map a real number data point a to the $\mathbb{Q}_{l_1,d}$ set, we use the function $g_{l_1,d}$, defined by the equation,

$$\begin{aligned} g_{l_1,d} &: \mathbb{R} \to \mathbb{Q}_{l_1,d} \\ g_{l_1,d}(a) &:= \arg \min_{q \in \mathbb{Q}_{l_1,d}} |a - q| \end{aligned} \tag{5.1.8}$$

Next, the quantized data is transformed into a set of integers through a one-to-one (bijective) mapping known as $f_{l_2,d}$, as outlined in [19]. The following mapping ensures that the quantized data is transformed into a subset of the message space $\mathbb{Z}_M$:

$$\begin{aligned} f_{l_2,d} &: \mathbb{Q}_{l_1,d} \to \mathbb{Z}_{2^{l_2}} \\ f_{l_2,d}(q) &:= 2^d q \bmod 2^{l_2} \end{aligned} \tag{5.1.9}$$

During the encryption process, integer plaintext messages from the set $Z_{2^{l_2}}$ are converted to ciphertexts, which can be decrypted back into the same set $Z_{2^{l_2}}$. To recover the original data from the set $\mathbb{Q}_{l_1,d}$, an inverse mapping, denoted as $f^{-1}_{l_2,d}$, is defined as follows:

$$f^{-1}_{l_2,d} : \mathbb{Z}_{2^{l_2}} \to \mathbb{Q}_{l_1,d} \tag{5.1.10}$$

$$f^{-1}_{l_2,d}(m) := \frac{1}{2^d} \begin{cases} m - 2^{l_2} & \text{if } m \geq 2^{l_2 - 1} \\ m & \text{otherwise} \end{cases} \tag{5.1.11}$$

5.1.3 Development of the encrypted distributed control architectures with state estimation

In this section, we describe the design and formulation of the encrypted distributed control architectures, both encrypted sequential and iterative distributed LMPCs with state estimation, provide additional details on the extended Luenberger observer.

5.1.3.1 Design of the encrypted sequential distributed LMPC

The control architecture of the encrypted sequential distributed LMPC is depicted in Figure 5.1.1. In a sequential distributed framework involving various LMPCs, communication is unidirectional. Specifically, the optimal control trajectory derived from solving the optimization problem for one LMPC is transmitted to another LMPC. This information is subsequently utilized by the receiving LMPC to proceed with its own optimization problem. The control strategy adheres to the following sequence of steps:

1. At time $t = t_k$, where k represents the sampling instance, signals $y(t_k)$ from sensors are encrypted to ciphertext c using the public key and transmitted to each control subsystem, within its respective edge computing unit.

2. Within each unit, the encrypted signals are decrypted using the private key, and the quantized states $\hat{y}(t_k)$ are used by the state estimator along with the control inputs computed at the previous sampling instance $u(t_{k-1})$ to estimate the current value of the states $\bar{x}(t_k)$.

3. The LMPC of the N_{sys}^{th} subsystem evaluates the optimal control trajectory $u^*_{N_{sys}}$ using the

estimated states $\bar{x}$ at $t = t_k$, and the stabilizing control law for the other $N_{sys} - 1$ subsystems, encrypts the control action of the first sampling period $u^*_{N_{sys}}(t_k)$ using the public key, transmits the ciphertext to the corresponding actuator, and transmits the entire optimal trajectory $u^*_{N_{sys}}(t|t_k)$, $t \in [t_k, t_{k+N})$ to the $N_{sys} - 1^{\text{th}}$ LMPC through the Ethernet crossover cable connection established between the different computing units.

4. The $N_{sys} - 1^{\text{th}}$ LMPC receives the entire optimal trajectory of the N^{th}_{sys} LMPC and evaluates the optimal trajectory $u_{N_{sys}-1}$ using the estimated states $\bar{x}(t_k)$ and the optimal input trajectory of the N^{th}_{sys} subsystem. It assumes the stabilizing control law for the remaining $N_{sys} - 2$ subsystems. It then encrypts the optimal trajectory for its respective subsystem over the next sampling period using the public key and transmits the complete optimal trajectory of subsystems N_{sys} and $N_{sys} - 1$ to the $N_{sys} - 2^{\text{th}}$ LMPC.

5. This same process is repeated up to the 1^{st} LMPC, which receives the optimal control input trajectories of all the other subsystems and computes its own optimal trajectory using the estimated states $\bar{x}(t_k)$ and the optimal control input trajectories of all the other subsystems.

6. At the actuator, the ciphertext $\acute{c}$ is decrypted to the quantized input $\hat{u}(t_k)$ using the private key, which is then applied to the process.

Formulation of the optimization problem, its constraints, and additional details of the encrypted sequential LMPC is presented in Section 5.1.3.4.

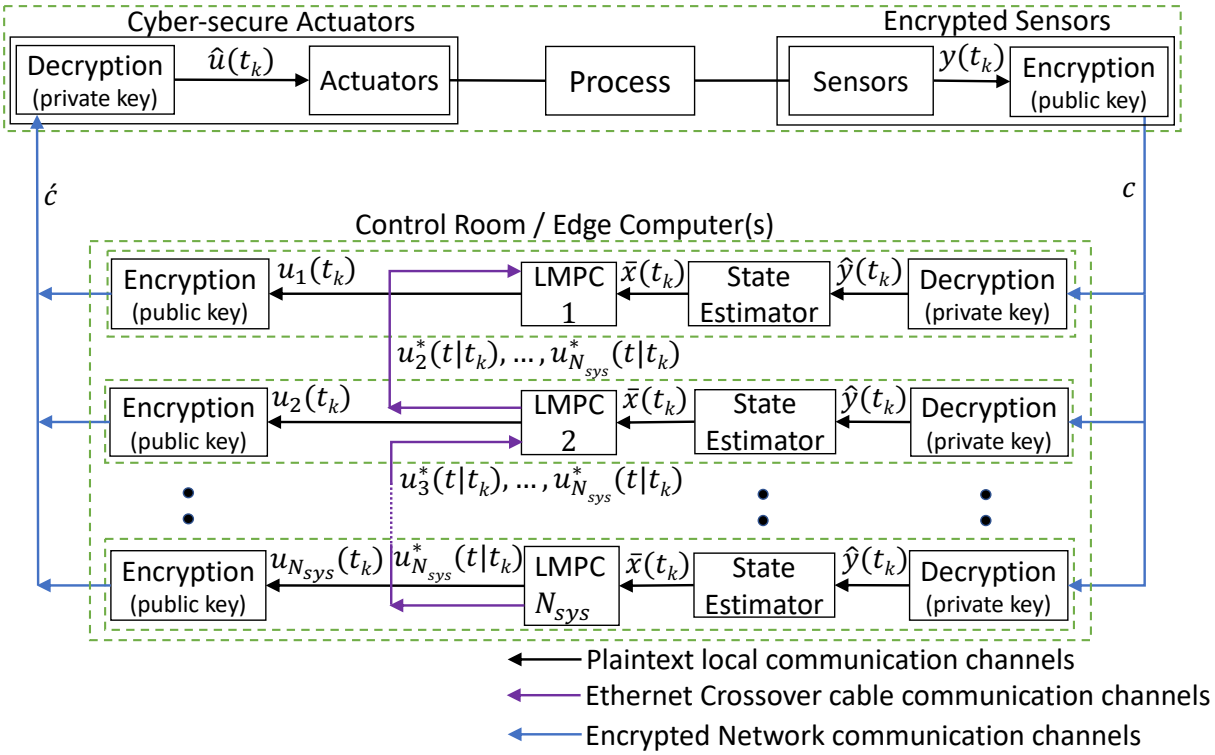

Figure 5.1.1: Illustration of the encrypted sequential distributed control structure.

5.1.3.2 Design of the encrypted iterative distributed LMPC

The control architecture of the encrypted iterative distributed LMPC is depicted in Figure 5.1.2. In this framework, all controllers communicate with each other to cooperatively optimize the control actions. The controllers solve their respective optimization problems independently within a parallel framework, and solutions for each control problem are exchanged at the end of each iteration. The control strategy adheres to the subsequent sequence of steps:

1. At time $t = t_k$, where k represents the sampling instance, signals $y(t_k)$ from sensors are encrypted to ciphertext c using the public key and transmitted to each control subsystem, within its respective edge computing unit.

2. Within each unit, the encrypted signals are decrypted using the private key, and the quantized

states $\hat{y}(t_k)$ are used by the state estimator along with the control inputs computed at the previous sampling instance $u(t_{k-1})$ to estimate the current value of all the system states $\bar{x}(t_k)$.

3. At iteration $z = 1$, the k^{th} LMPC in the k^{th} subsystem evaluates optimal control input trajectories $u_k^*(t)$, using the estimated states $\bar{x}(t_k)$, and assuming $u_j(t) = \Phi_j(\bar{x}(t))$ where $j \in \{1, \ldots, N_{sys}\}, \quad j \neq k$. At the end of the first iteration, each subsystem transmits its complete optimal control input trajectory to all N_{sys} subsystems through the Ethernet crossover cable connection established between the different control subsystems.

4. At iteration $z = 2$, the k^{th} LMPC in the k^{th} subsystem re-evaluates optimal control input trajectories $u_k^*(t)$ using the estimated states $\bar{x}(t_k)$, and the optimal control input trajectories $u_j^*(t)$ where $j \in \{1, \ldots, N_{sys}\}, \quad j \neq k$. At the end of the second iteration, each subsystem transmits its complete optimal control input trajectory to all N_{sys} subsystems. This process is continued until a termination criterion is satisfied. The termination criterion can be either that the number of iterations, denoted as z, must not exceed the maximum number of iterations, denoted as $z_{\max}$, or that the difference in the value of the cost function between two consecutive iterations is smaller than a threshold value.

5. After the termination criterion is satisfied, each LMPC encrypts its control input corresponding to the lowest cost function over the next sampling period (using the public key), and the encrypted ciphertext is transmitted to the corresponding actuators of that particular subsystem.

6. At the actuator, the ciphertext $\acute{c}$ is decrypted to the quantized input $\hat{u}(t_k)$ using the private key, which is then applied to the process.

Formulation of the optimization problem, its constraints, and additional details for the iterative encrypted DMPC are presented in Section 5.1.3.5.

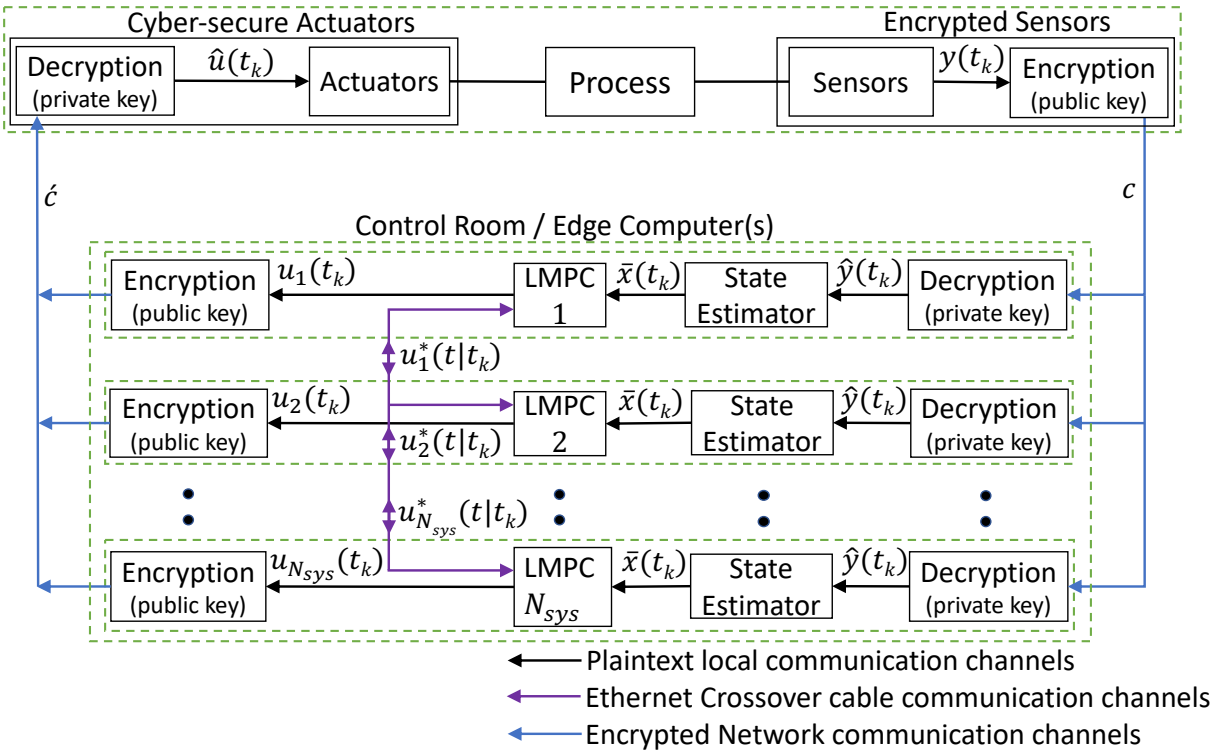

Figure 5.1.2: Illustration of the encrypted iterative distributed control structure.

Remark 5.1.4. *In the closed-loop block diagrams shown in Figure* 5.1.1 *and Figure* 5.1.2*, Ethernet crossover cable connections facilitate communication between the computing units of different subsystems. This setup assumes a secure edge computer(s) within a protected control room, where encrypted signals from sensors at the process site are received and from where encrypted control inputs are transmitted to the actuators. However, communication between subsystems responsible for computing control inputs remains unencrypted. The rationale behind this decision is to mini-*

mize the communication overhead due to encryption–decryption in the control system. Complete control input trajectories must be communicated multiple times within a single sampling period in the case of iterative DMPC. Encrypting and decrypting these trajectories repeatedly within a single sampling period may not be feasible, particularly for very large systems. Such repetition could lead to increased communication overhead. Since the primary objective of a DMPC is to distribute the optimization problem among separate computing units and solve each one effectively, the assumption of having all responsible edge computing units in a secure room with secure cable connections between them is reasonable. Alternatively, the option to encrypt and decrypt inputs could be considered if the initial arrangement is not achievable. Further insights into the communication and computational implications associated with encryption and decryption are available in [39].

The closed-loop design of Figure 5.1.1 and Figure 5.1.2 introduces two sources of error: one from state quantization in the sensor–controller link and another from control input quantization in the controller–actuator link. These errors are bounded by:

$$|y(t_k) - \hat{y}(t_k)| \leq 2^{-d-1} \tag{5.1.12a}$$

$$|u(t_k) - \hat{u}(t_k)| \leq 2^{-d-1} \tag{5.1.12b}$$

The state estimator, as expressed in Eq. (5.1.2), can be written as a function $\phi(\bar{x}, y, u)$. An additional error arises in the applied control input, as the state estimator receives $\hat{y}$ instead of the true state y to estimate all the system states. Using the local Lipschitz property, this error will be

confined by the underlying equation, where $L_1' > 0$:

$$|\phi(\bar{x}, \hat{y}, u) - \phi(\bar{x}, y, u)| \leq L_1' |\hat{y} - y| \leq L_1' 2^{-d-1} \tag{5.1.13}$$

Remark 5.1.5. *A quantization error occurs when the value to be quantized does not precisely match a member of the set $\mathbb{Q}_{l_1,d}$. The elements in this set are spaced apart by 2^{-d}, which represents the resolution of the set. Let us assume the value to be quantized is denoted as a, and it falls within the range of b to $b + 2^{-d}$. If the absolute difference between a and b is smaller than the difference between a and $b + 2^{-d}$, a is assigned to the value b. Otherwise, it is assigned to the value $b + 2^{-d}$. Consequently, the maximum potential discrepancy between the actual and quantized values is half of the resolution, which is equal to 2^{-d-1}. This limitation also implies that a greater value of d would lead to a reduced quantization error.*

5.1.3.3 Extended Luenberger observer-based state estimation

An extended Luenberger observer (ELO) is employed to estimate all the states of the nonlinear system, as detailed in Eq. (5.1.1). This estimation process relies on noisy partial state feedback obtained from sensors after decryption. Consequently, each subsystem's computing unit integrates an ELO, initializing the LMPC model of each subsystem with a complete state estimate (through the ELO) denoted as $\bar{x}$. In the design of Eq. (5.1.2), the observer necessitates a process model of the nonlinear system. Interestingly, the LMPC model can be extended and utilized within the observer, or vice versa. This dual utilization presents an effective approach for large-scale processes, reducing the number of required measured states through the ELO, and enhancing closed-loop

performance and ensuring stability through the constraints of a Lyapunov-based MPC.

The typical sequence of actions executed by an ELO within the computing unit assigned to compute control inputs for a particular subsystem is as follows:

1. At time $t = t_k$, where k is the sampling instance, the ELO process model is initialized using all the estimated states of the system at the previous sampling instance $\bar{x}(t_{k-1})$, and all the control inputs computed at the previous sampling instance $u(t_{k-1})$.

2. The ELO process model predicts the state at the next integration time step $\bar{x}(t_{k-1}+h_c)$, where h_c represents the integration time step. A correction term $h_c \times K(\hat{y}(t_k) - h(\bar{x}(t_{k-1} + h_c)))$ is added to the estimated state $\bar{x}(t_{k-1} + h_c)$. Here, $\hat{y}(t_k)$ is the quantized measured state vector after decryption at time $t = t_k$.

3. The above step is reiterated Δ/h_c times, with Δ representing the sampling period, in order to compute the final estimated state at t_k, denoted as $\bar{x}(t_k)$. It is important to note that the control input $u(t_{k-1})$ remains constant within a single sampling period, as it is applied in a sample-and-hold manner and does not undergo any change during this interval.

During the initial sampling period, denoted as t_0, we make the assumption that the control inputs and the initial estimated states are set to their steady-state values. This assumption is necessary as no prior data is accessible for this specific sampling instance.

Remark 5.1.6. *The procedure outlined in Remark 5.1.1 involves linearizing the nonlinear system described in Eq.* (5.1.1) *around its steady state. The observer gains are adjusted in such a way that the matrix $A - KL$ in Eq.* (5.1.4) *possesses eigenvalues with negative real components. However, since the observer is intended to be applied to a nonlinear system, further fine-tuning of the*

gains might be required. To achieve this, multiple simulations of the observer integrated within the nonlinear system, along with the state feedback controller, are conducted. These simulations encompass random initial conditions within the set Ω_ρ. During this process, the observer gains are refined to ensure that the error $e = x - \bar{x}$ tends to zero or a sufficiently small threshold, within a finite number of iterations for each randomly initialized simulation. Each iteration corresponds to a sampling period. Furthermore, with these newly fine-tuned observer gains, it is ensured that the matrix $A - KL$ continues to possess eigenvalues with negative real components. This adjustment is particularly necessary for nonlinear systems because the assumptions and properties of a linear system cannot be directly extrapolated to nonlinear systems.

5.1.3.4 Encrypted sequential distributed LMPC

In order to mitigate the computational time and complexity associated with a centralized control problem, especially in the context of large-scale systems featuring multiple states and control inputs, we propose the establishment of a sequential distributed LMPC system, where the opti-

mization problem for the j^{th} LMPC is delineated as follows:

$$\mathcal{J} = \min_{u_{d_j} \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), \Phi_m(\tilde{x}(t)), u_{d_n}(t))\, \mathrm{d}t,$$

$$\text{where } m = 1, \ldots, j-1 \text{ and } n = j, \ldots, N_{sys} \tag{5.1.14a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F(\tilde{x}(t), \Phi_m(\tilde{x}(t)), u_{d_n}(t)) \tag{5.1.14b}$$

$$\dot{\bar{x}}(t) = F(\bar{x}(t), \Phi_m(\bar{x}(t)), u_{d_n}(t)) + K(\hat{y}(t) - h(\bar{x}(t))) \tag{5.1.14c}$$

$$u_{d_j}(t) \in U_j, \ \forall\, t \in [t_k, t_{k+N}) \tag{5.1.14d}$$

$$\tilde{x}(t_k) = \bar{x}(t_k) \tag{5.1.14e}$$

$$\dot{V}(\bar{x}(t_k), \Phi_m(\bar{x}(t_k)), u_{d_n}(t_k)) \leq \dot{V}(\bar{x}(t_k), \Phi_m(\bar{x}(t_k)), \Phi_n(\bar{x}(t_k)))$$

$$\text{if } \bar{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{5.1.14f}$$

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

$$\text{if } \bar{x}(t_k) \in \Omega_{\rho_{\min}} \tag{5.1.14g}$$

At time $t = t_k$, where k represents the sampling instance, the ELO in the computing unit corresponding to the j^{th} LMPC decrypts the ciphertext c to receive the quantized state measurements $\hat{y}(t_k)$. The ELO uses these along with the computed control inputs at the previous sampling instance, i.e., $\Phi_m(\bar{x}(t_{k-1}))$ and $u_{d_n}(t_{k-1})$, where $m = 1, \ldots, j-1$ and $n = j, \ldots, N_{sys}$, and the estimated states at the previous sampling instance $\bar{x}(t_{k-1})$ to predict the states at the current sampling instance, $\bar{x}(t_k)$, through Eq. (5.1.14c). The j^{th} LMPC then receives the complete state estimate $\bar{x}(t_k)$ from the ELO, but only computes the control input of its subsystem, u_{d_j}, which is to be applied by the corresponding actuators. It assumes the stabilizing control law for control

inputs of subsystems 1 to $j-1$, and receives the optimal control input trajectories $u_{d_{n'}}$ from the remaining n' subsystems where $n' = j+1, \ldots, N_{sys}$. $\tilde{x}$ represents the predicted state trajectory of the process model of the j^{th} LMPC. The estimated states, $\bar{x}$, serve as the initial conditions for the LMPC process model to predict the state trajectory as per Eq. (5.1.14b), which is used to integrate the cost function of Eq. (5.1.14a) to calculate optimized control inputs, $u^*_{d_j}(t)$, $t \in [t_k, t_{k+N})$, for the entire prediction horizon. However, the LMPC transmits only the first input of this sequence, $u^*_{d_j}(t_k)$ to the actuator for application to the system within the interval $t \in [t_k, t_{k+1})$ and transmits the entire control input trajectory $u^*_{d_j}(t)$ along with the received control input trajectory, $u^*_{d_{n'}}$ where $n' = j+1, \ldots, N_{sys}$ to the $j-1^{\text{th}}$ LMPC. This process is repeated at each sampling period. N represents the number of sampling periods within the prediction horizon. Eq. (5.1.14d) represents the constraints imposed on the control inputs, and Eq. (5.1.14e) uses the quantized states to initialize the plant model described in Eq. (5.1.14b). The Lyapunov constraint in Eq. (5.1.14f) ensures that, if the state $\bar{x}(t_k)$ at time t_k lies within the set $\Omega_\rho \setminus \Omega_{\rho_{\min}}$, where $\rho_{\min}$ represents a level set of V in proximity to the origin, the time-derivative of the control Lyapunov function of the closed-loop subsystem j under the j^{th} LMPC, and stabilizing control law for the other control inputs, is less than or equal to the time-derivative of the control Lyapunov function when the subsystem is controlled by the stabilizing controller $\Phi(\bar{x})$. When the closed-loop state $\bar{x}(t_k)$ enters $\Omega_{\rho_{\min}}$, the constraint of Eq. (5.1.14g) ensures that this state remains within $\Omega_{\rho_{\min}}$.

Remark 5.1.7. *Within the proposed framework, a secure edge computer receives the encrypted partial state feedback. This computer then decrypts the received encrypted partial state feedback and employs the extended Luenberger observer within the same unit to compute all states, using the*

quantized partial state feedback values. Following this process, the LMPC utilizes the estimated states received from the observer, all within the same computing unit. Since these operations occur internally in the same unit, they are not encrypted. However, the control input computed by the LMPC, to be sent to and applied by the actuator, is encrypted before transmission. This configuration ensures that all wireless networked communications remain encrypted, thereby enhancing the cybersecurity of the control system.

Remark 5.1.8. *Although state constraints have not been explicitly utilized in our LMPC formulations (only input constraints are considered), they can still be integrated if required for both the LMPC and extended Luenberger observer. In the case of the ELO, one feasible approach could involve utilizing the estimated states from the observer and subsequently applying a post-processing technique (like modifying the observer gain and re-running the observer) to ensure adherence to the defined constraints. Future research can be conducted to identify other methods to account for state constraints within the observer.*

5.1.3.5 Encrypted iterative distributed LMPC

An alternative approach to a sequential DMPC is the iterative DMPC, in which the controllers responsible for computing control inputs for each subsystem of the overall process share their control inputs at the end of each iteration until a termination criterion is met. The optimization problem for the j^{th} LMPC within the iterative distributed LMPC structure, for the first iteration,

$z = 1$, is described as follows:

$$\mathcal{J} = \min_{u_{d_j} \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), \Phi_m(\tilde{x}(t)), u_{d_j}(t))\, \mathrm{d}t,$$

$$\text{where } m = 1, \ldots, N_{sys} \text{ and } m \neq j \tag{5.1.15a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F(\tilde{x}(t), \Phi_m(\tilde{x}(t)), u_{d_j}(t)) \tag{5.1.15b}$$

$$\dot{\bar{x}}(t) = F(\bar{x}(t), u_{d_m}(t), u_{d_j}(t)) + K(\hat{y}(t) - h(\bar{x}(t))) \tag{5.1.15c}$$

$$u_{d_j}(t) \in U_j, \ \forall\, t \in [t_k, t_{k+N}) \tag{5.1.15d}$$

$$\tilde{x}(t_k) = \bar{x}(t_k) \tag{5.1.15e}$$

$$\dot{V}(\bar{x}(t_k), \Phi_m(\bar{x}(t_k)), u_{d_j}(t_k)) \leq \dot{V}(\bar{x}(t_k), \Phi_m(\bar{x}(t_k)), \Phi_j(\bar{x}(t_k))),$$

$$\text{if } \bar{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{5.1.15f}$$

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

$$\text{if } \bar{x}(t_k) \in \Omega_{\rho_{\min}} \tag{5.1.15g}$$

At the iteration $z > 1$ following the exchange of the optimized input trajectories $u^*_{d_m}(t)$ with the rest of the LMPCs, the optimization problem of j^{th} LMPC is modified as follows:

$$\mathcal{J} = \min_{u_{d_j} \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), u_{d_m}(t), u_{d_j}(t))\, \mathrm{d}t,$$

$$\text{where } m = 1, \ldots, N_{sys} \text{ and } m \neq j \tag{5.1.16a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F(\tilde{x}(t), u_{d_m}(t), u_{d_j}(t)) \tag{5.1.16b}$$

$$\dot{\hat{x}}(t) = F(\bar{x}(t), u_{d_m}(t), u_{d_j}(t)) + K(\hat{y}(t) - h(\bar{x}(t))) \tag{5.1.16c}$$

$$u_{d_j}(t) \in U_j, \ \forall\, t \in [t_k, t_{k+N}) \tag{5.1.16d}$$

$$\tilde{x}(t_k) = \bar{x}(t_k) \tag{5.1.16e}$$

$$\dot{V}(\bar{x}(t_k), u_{d_m}(t_k), u_{d_j}(t_k)) \leq \dot{V}(\bar{x}(t_k), \Phi_m(\bar{x}(t_k)), \Phi_j(\hat{x}(t_k))),$$

$$\text{if } \bar{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{5.1.16f}$$

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

$$\text{if } \bar{x}(t_k) \in \Omega_{\rho_{\min}} \tag{5.1.16g}$$

The j^{th} LMPC receives the complete state estimate $\bar{x}(t_k)$ from the ELO, but only computes the control input of its specific subsystem, denoted as u_{d_j}, which is to be applied by the corresponding actuators. Initially, for the first iteration, $z = 1$, it assumes the stabilizing control law for control inputs of m subsystems, where $m = 1, \ldots, N_{sys}$, and $m \neq j$. Subsequently, for iterations $z > 1$, the j^{th} LMPC transmits its computed control input at the previous iteration to all other LMPCs, and receives the control inputs computed by all other LMPCs at the previous iteration over the entire prediction horizon. The j^{th} LMPC then recalculates the control inputs for its respective

subsystem, assuming the received control input trajectories for the other subsystems. At the end of the current iteration, it transmits the updated control input trajectory of its subsystem to the other subsystems. This is repeated until a termination criterion is satisfied. The formulation of the optimization problems presented in Eq. (5.1.15), and Eq. (5.1.16) is very similar to Eq. (5.1.14), which was elaborated in detail in Section 5.1.3.4.

Remark 5.1.9. *In the context of the stability analysis for the introduced encrypted distributed LMPC architectures in this study, the bounds related to encryption-induced errors have been established in Section* 5.1.3.2*. Additionally, each LMPC in the distributed structure incorporates a constraint stipulating that the value of the time-derivative of the control Lyapunov function under the LMPC should be more negative than that of the observer-based stabilizing control law. A comprehensive stability analysis has previously been conducted for a nonlinear centralized encrypted system in [81]. Building on this foundation, a similar stability analysis can be carried out for the encrypted distributed LMPC, incorporating an observer. It is important to note that, given our assumption of an observer-based stabilizing control law, the stability analysis is simplified and does not require elaborate demonstration; thus, it has been omitted.*

5.1.4 Application to a nonlinear chemical process network operating at an unstable steady state

This section demonstrates the proposed encrypted distributed control architectures, both sequential and iterative distributed LMPCs with state estimation, on a nonlinear chemical process network with noisy partial state feedback, operating at an unstable steady state. A nonlinear dynamical

model based on first-principles modeling fundamentals is developed for the state estimator and the LMPCs. Guidelines are established to implement the encrypted distributed LMPC systems in any nonlinear process with partial state feedback. We then conduct closed-loop simulations, employing the distributed LMPCs with state estimators, and analyze the results.

5.1.4.1 Process description and model development

The process considered is the synthesis of ethylbenzene (EB) by reacting ethylene (E) and benzene (B) within two non-isothermal, well-mixed continuous stirred tank reactors (CSTRs) as depicted in Figure 5.1.3. The primary reaction, termed as "primary", is characterized as a second-order, exothermic, and irreversible reaction, in conjunction with two supplementary side reactions. The chemical reactions taking place are articulated as follows:

$$C_2H_4 + C_6H_6 \rightarrow C_8H_{10} \qquad \text{(primary)} \tag{5.1.17a}$$

$$C_2H_4 + C_8H_{10} \rightarrow C_{10}H_{14} \tag{5.1.17b}$$

$$C_6H_6 + C_{10}H_{14} \rightarrow 2C_8H_{10} \tag{5.1.17c}$$

Details of the steady-state values and model parameter values can be obtained from [39]. The dynamic model of the first CSTR is described by the following mass and energy balance equations:

$$\dot{C}_{E_1} = \frac{F_1 C_{Eo_1} - F_{out_1} C_{E_1}}{V_1} - r_{1,1} - r_{1,2} \tag{5.1.18a}$$

$$\dot{C}_{B_1} = \frac{F_1 C_{Bo_1} - F_{out_1} C_{B_1}}{V_1} r_{1,1} - r_{1,3} \tag{5.1.18b}$$

$$\dot{C}_{EB_1} = \frac{-F_{out_1} C_{EB_1}}{V_1} + r_{1,1} - r_{1,2} + 2r_{1,3} \tag{5.1.18c}$$

$$\dot{C}_{DEB_1} = \frac{-F_{out_1} C_{DEB_1}}{V_1} + r_{1,2} - r_{1,3} \tag{5.1.18d}$$

$$\dot{T}_1 = \frac{T_{1_o} F_1 - T_1 F_{out_1}}{V_1} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_1 C_p} r_{1,j} + \frac{Q_1}{\rho_1 C_p V_1} \tag{5.1.18e}$$

The dynamic model of the second CSTR is represented by the following equations:

$$\dot{C}_{E_2} = \frac{F_2 C_{Eo_2} + F_{out_1} C_{E_1}}{V_2} - \frac{F_{out_2} C_{E_2}}{V_2} - r_{2,1} - r_{2,2} \tag{5.1.19a}$$

$$\dot{C}_{B_2} = \frac{F_2 C_{Bo_2} + F_{out_1} C_{B_1}}{V_2} - \frac{F_{out_2} C_{B_2}}{V_2} - r_{2,1} - r_{2,3} \tag{5.1.19b}$$

$$\dot{C}_{EB_2} = \frac{F_{out_1} C_{EB_1} - F_{out_2} C_{EB_2}}{V_2} + r_{2,1} - r_{2,2} + 2r_{2,3} \tag{5.1.19c}$$

$$\dot{C}_{DEB_2} = \frac{F_{out_1} C_{DEB_1} - F_{out_2} C_{DEB_2}}{V_2} + r_{2,2} - r_{2,3} \tag{5.1.19d}$$

$$\dot{T}_2 = \frac{T_{2_o} F_2 + T_1 F_{out_1} - T_2 F_{out_2}}{V_2} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_2 C_p} r_{2,j} + \frac{Q_2}{\rho_2 C_p V_2} \tag{5.1.19e}$$

where the reaction rates are calculated by the following expressions:

$$r_{i,1} = k_1 \mathrm{e}^{\frac{-E_1}{RT_i}} C_{E_i} C_{B_i} \tag{5.1.20a}$$

$$r_{i,2} = k_2 \mathrm{e}^{\frac{-E_2}{RT_i}} C_{E_i} C_{EB_i} \tag{5.1.20b}$$

$$r_{i,3} = k_3 \mathrm{e}^{\frac{-E_3}{RT_i}} C_{DEB_i} C_{B_i} \tag{5.1.20c}$$

where $i = \{1, 2\}$ is the reactor index. The state variables are the concentration of ethylene, benzene, ethylbenzene, di-ethylbenzene, and the reactor temperature for each CSTR in deviation terms, that is: $x^\top = [C_{E_1} - C_{E_{1s}},\ C_{B_1} - C_{B_{1s}},\ C_{EB_1} - C_{EB_{1s}},\ C_{DEB_1} - C_{DEB_{1s}},\ T_1 - T_{1s},\ C_{E_2} - C_{E_{2s}},\ C_{B_2} - C_{B_{2s}},\ C_{EB_2} - C_{EB_{2s}},\ C_{DEB_2} - C_{DEB_{2s}},\ T_2 - T_{2s}]$. The subscript "$s$" denotes the steady-state value. The desired product, ethyl benzene, and the CSTR temperature are the measured states corresponding to $y^\top = [C_{EB_1} - C_{EB_{1s}},\ T_1 - T_{1s},\ C_{EB_2} - C_{EB_{2s}},\ T_2 - T_{2s}]$. The measured states are visually represented in blue in Figure 5.1.3. In contrast, the remaining states that are not measured are depicted in red within the same figure. Bounded white Gaussian noise is added to the measured states of both CSTRs. The mean of the noise is zero for both states, and the standard deviation is $0.003\,\mathrm{kmol/m^3}$ for the measured concentration of ethylbenzene and $0.15\,\mathrm{K}$ for the measured CSTR temperature, in each CSTR. The noise is bounded by the closed sets [-0.01, 0.01] $\mathrm{kmol/m^3}$ and [-0.5, 0.5] K for the measured ethylbenzene concentration and temperature states, respectively.

The rate of heat removal for the two reactors $[Q_1 - Q_{1s},\ Q_2 - Q_{2s}]$ and inlet feed concentrations for each reactor, $[C_{E_{o1}} - C_{E_{o1s}}, C_{B_{o1}} - C_{B_{o1s}}, C_{E_{o2}} - C_{E_{o2s}}, C_{B_{o2}} - C_{B_{o2s}}]$, are the manipulated inputs of the nonlinear system. These inputs are bounded by the closed sets, $[-10^4, 2 \times 10^3]$ kW, $[-1.5 \times 10^4, 5 \times 10^3]$ kW, $[-2.5, 2.5]$ $\mathrm{kmol/m^3}$, $[-2.5, 2.5]$ $\mathrm{kmol/m^3}$, $[-3, 3]$ $\mathrm{kmol/m^3}$, and $[-3, 3]$ $\mathrm{kmol/m^3}$, respectively. To assess the stability of the selected equilibrium state, an open-loop simulation was conducted. During this simulation, the control inputs remained fixed at their equilibrium values, and the initial conditions of the system were set near the operating equilibrium point, within the region $\Omega_{\rho_{\min}}$. After a finite period of time, the system's states departed from the stability region Ω_ρ, and eventually converged to an entirely different equilibrium state. This

transition signifies the instability of the initial equilibrium. The rationale for choosing this particular state was its capability to achieve a significantly high steady-state concentration of the desired product, ethyl benzene of $4.22\,\mathrm{kmol/m^3}$, at the outlet of reactor 2, under reasonable operating conditions.

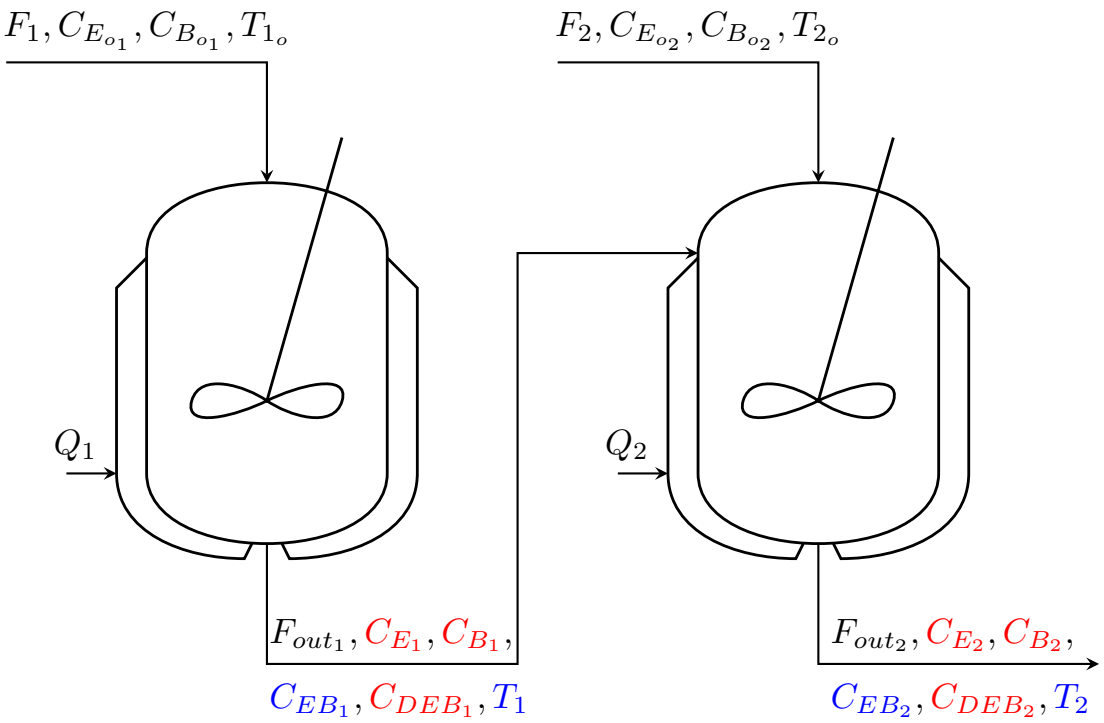

Figure 5.1.3: Process schematic featuring two CSTRs connected in series.

The overall control of the system was partitioned into two LMPCs. Both LMPCs utilized a first-principles-based model, and received the estimated states $\bar{x}$ from the ELO. Further, LMPC 1 optimizes the control inputs $u_1 = [C_{E_{o1}} - C_{E_{o1s}}, C_{B_{o1}} - C_{B_{o1s}}, Q_1 - Q_{1s}]^\top$, while LMPC 2 optimizes the control inputs $u_2 = [C_{E_{o2}} - C_{E_{o2s}}, C_{B_{o2}} - C_{B_{o2s}}, Q_2 - Q_{2s}]^\top$. Thus, the partitioning of the overall systems is done such that LMPC 1 manipulates all the control inputs of CSTR 1, while LMPC 2 manipulates all the control inputs of CSTR 2. In case of the sequential distributed LMPC system, LMPC 2 assumes the control inputs for LMPC 1 as per the stabilizing control

law, and accordingly computes the optimized control inputs for its subsystem, CSTR 2. It then transmits the control input trajectory over the complete prediction horizon to LMPC 1, which uses this information to compute the control inputs of its respective subsystem, CSTR 1. On the other hand, in the iterative distributed LMPC system, in the first iteration, both LMPCs compute the control inputs of their respective subsystems, assuming the stabilizing control law for the inputs of the other subsystem. At the second iteration, both LMPCs, share the control input trajectory over the prediction horizon, computed for their respective subsystems with each other. Based on the information received about the control inputs of the other subsystem, both LMPCs recompute the optimized control inputs of their respective subsystem. This exchange of information goes on until a termination criterion is satisfied. In the example demonstrated in this section, we have used a termination criterion of 2 iterations for the iterative distributed LMPC. The control objective is to operate both CSTRs at their unstable equilibrium point through the encrypted distributed control schemes, sequential and iterative, employing quantized partial state feedback with sensor noise for computation of the required control inputs.

5.1.4.2 Encrypting the distributed control architectures

Prior to integrating encryption and decryption into a process, the process of parameter selection, specifically involving the variables d, l_1, and l_2, takes place. By considering the extreme feasible states and inputs, the integer bit count $l_1 - d$ is determined. The upper limit within the set $\mathbb{Q}_{l_1,d}$ is calculated using the formula $2^{l_1-d-1} - 2^{-d}$, while the lower limit is established as -2^{l_1-d-1}. The selection of the quantization parameter d, which represents the fractional bit count, depends on the desired level of precision and the range of state and input values. Additionally, l_2 is chosen

to be greater than l_1. In the context of the example presented in this section, a value of 16 is determined for $l_1 - d$, which in turn determines the values of l_1 and d. Within the set $\mathbb{Q}_{l_1,d}$, rational numbers are spaced apart by a resolution of 2^{-d}. For simulation purposes, we have opted for a value of $d = 8$. With $d = 8$, l_1 is set at 24, and l_2 is selected as 30. The implementation of the Paillier Encryption procedure is carried out using the Python "phe" module, specifically PythonPaillier [21]. For solving the multi-constrained, non-convex optimization problem within the LMPCs operating within the distributed control framework, we utilize the Python module from the IPOPT software [83].

While deciding the sampling time (Δ) for an encrypted distributed system with state estimation, it is crucial to ensure that it exceeds the total time required for encryption–decryption of the states and control inputs, time required by the state estimator to estimate the states, and the time needed to compute all the control inputs at each sampling instance for the considered quantization parameter d. Encryption–decryption and state estimation is performed in parallel between different edge computing units. Hence, we select the maximum time from all the different subsystems across all sampling instances. As control input information is exchanged, control input computation time is the total time needed to compute all the control inputs, and not just inputs for a particular subsystem. Hence, we select the maximum time taken to compute all the control inputs at any sampling instance. Mathematically,

$$\begin{aligned}\Delta > \max\left(\text{encryption–decryption time}\right)_j + \max\left(\text{State-estimation time}\right)_j \\ + \max\left(\text{Control input computation time}\right)\end{aligned} \tag{5.1.21}$$

where $j = \{1, \ldots, N_{sys}\}$ represents the control subsystem. Details on how the control input com-

putation time is calculated for the sequential and iterative DMPCs is provided in the next section. Considering the above criteria, the sampling time Δ is chosen as 30 seconds in the discussed example. In real-world scenarios, the state estimation computations in a process simulation (not the actual process) could take place on the specific type of computer intended for the actual usage of these calculations. The computational time across multiple sampling instances of the process simulation can be recorded, and the maximum duration among these instances can be chosen as the maximum state-estimation time. This same concept can be extended to obtain the maximum encryption-decryption time. Additionally, to account for precautionary measures, this value could be multiplied by a factor, such as 1.25.

To calculate the cost function of the LMPCs over the prediction horizon, an integration time step, $h_c = 10^{-2} \times \Delta$, is chosen. The positive definite matrix P in the control Lyapunov function $V = x^\top Px$ is selected as $\mathrm{diag}\,[200\ \ 200\ \ 400\ \ 1000\ \ 2.5\ \ 250\ \ 250\ \ 200\ \ 1000\ \ 0.5]$, from extensive simulations. The LMPCs employ a prediction horizon of $N = 2$ sampling periods. The stability criterion is defined as $\rho = 1800$, while $\rho_{\min} = 2$ is the smaller level set of the Lyapunov function where the state is desired to be confined. The weight matrix in the cost function of LMPCs is chosen as $Q = \mathrm{diag}\,[1000\ \ 1000\ \ 1500\ \ 5\ \ 8\ \ 1000\ \ 1000\ \ 3000\ \ 5\ \ 110]$, $R = \mathrm{diag}\,[2.1\ \ 1.95\ \ 1.5 \times 10^{-5}\ \ 10\ \ 10\ \ 0.5 \times 10^{-4}]$. The cost function is defined as $L(x, u) = x^\top Qx + u^\top Ru$.

5.1.4.3 Simulation results of the encrypted distributed control architectures

The proposed encrypted distributed control architecture is applied to a nonlinear chemical process, and the control inputs are computed using partial state feedback with sensor noise. Figures 5.1.4 to 5.1.6 depict the results for the encrypted sequential distributed LMPC system with state estima-

tion from the first set of initial conditions, $x_0 = [-0.35\,\text{kmol/m}^3, -0.3\,\text{kmol/m}^3, 0.2\,\text{kmol/m}^3, 0\,\text{kmol/m}^3, -20\,\text{K}, 0.2\,\text{kmol/m}^3, 0.15\,\text{kmol/m}^3, -0.25\,\text{kmol/m}^3, 0\,\text{kmol/m}^3, -15\,\text{K}]^\top$.

Figures 5.1.7 to 5.1.9 depict the results for the encrypted iterative distributed LMPC system from the second set of initial conditions, $x_0 = [0.5\,\text{kmol/m}^3, 0.35\,\text{kmol/m}^3, -0.2\,\text{kmol/m}^3, 0\,\text{kmol/m}^3, 20\,\text{K}, 0.45\,\text{kmol/m}^3, 0.5\,\text{kmol/m}^3, -0.8\,\text{kmol/m}^3, 0\,\text{kmol/m}^3, -30\,\text{K}]^\top$.

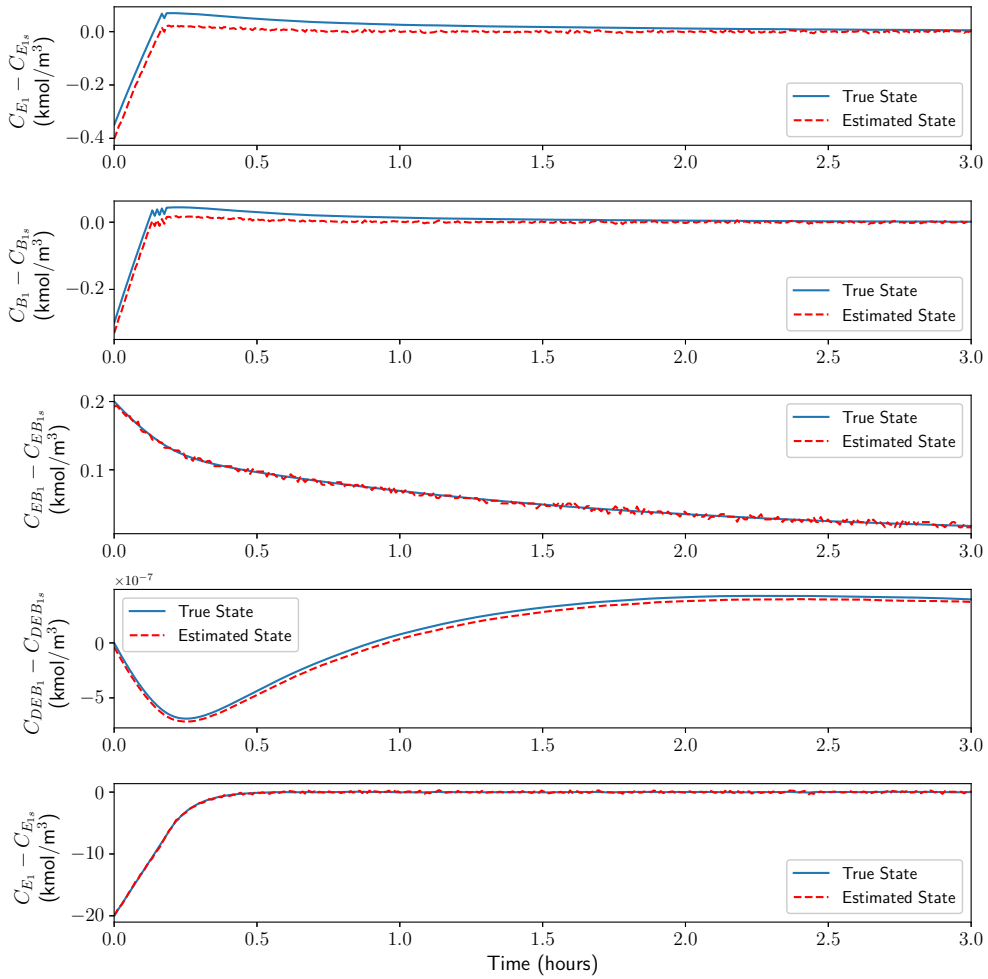

Figure 5.1.4: True state profiles (blue solid line) and estimated state profiles (red dashed line) of CSTR 1 under the encrypted sequential distributed LMPC framework for the first set of initial conditions.

In Figures 5.1.4, 5.1.5, 5.1.7 and 5.1.8, the blue solid line represents the true state value, while the red dashed line represents the state value estimated by the ELO. For both initial conditions, the

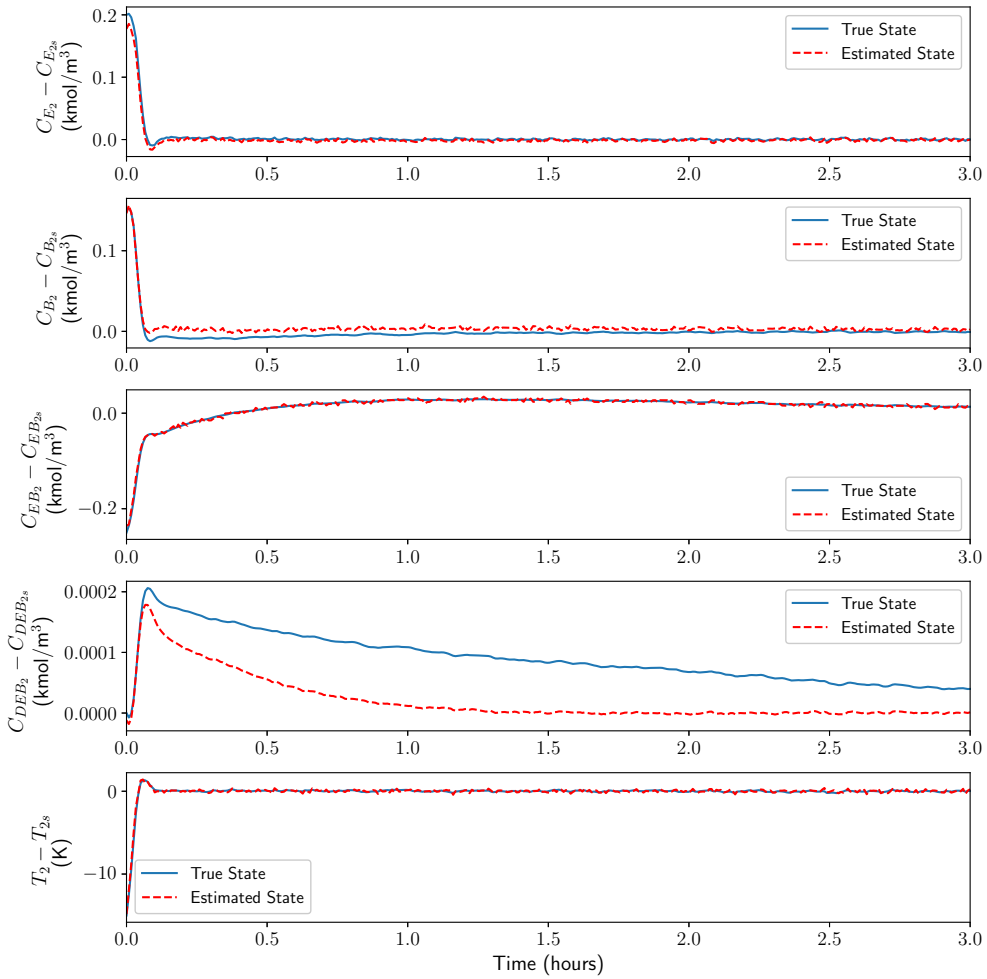

Figure 5.1.5: True state profiles (blue solid line) and estimated state profiles (red dashed line) of CSTR 2 under the encrypted sequential distributed LMPC framework for the first set of initial conditions.

state estimator (ELO) provides the distributed LMPCs with fairly accurate state estimates, using partial state feedback with sensor noise. Minor deviations between the estimated and predicted states are noticeable in Figure 5.1.5. These deviations can be attributed to the observer receiving partial state feedback with sensor noise. Additionally, errors stemming from quantization can also play a role in this discrepancy. However, it is essential to note that both sources of error are bounded, as previously indicated, resulting in the observed deviations being minor in nature. The distributed LMPCs successfully stabilize the system within the desired closed-loop stability region

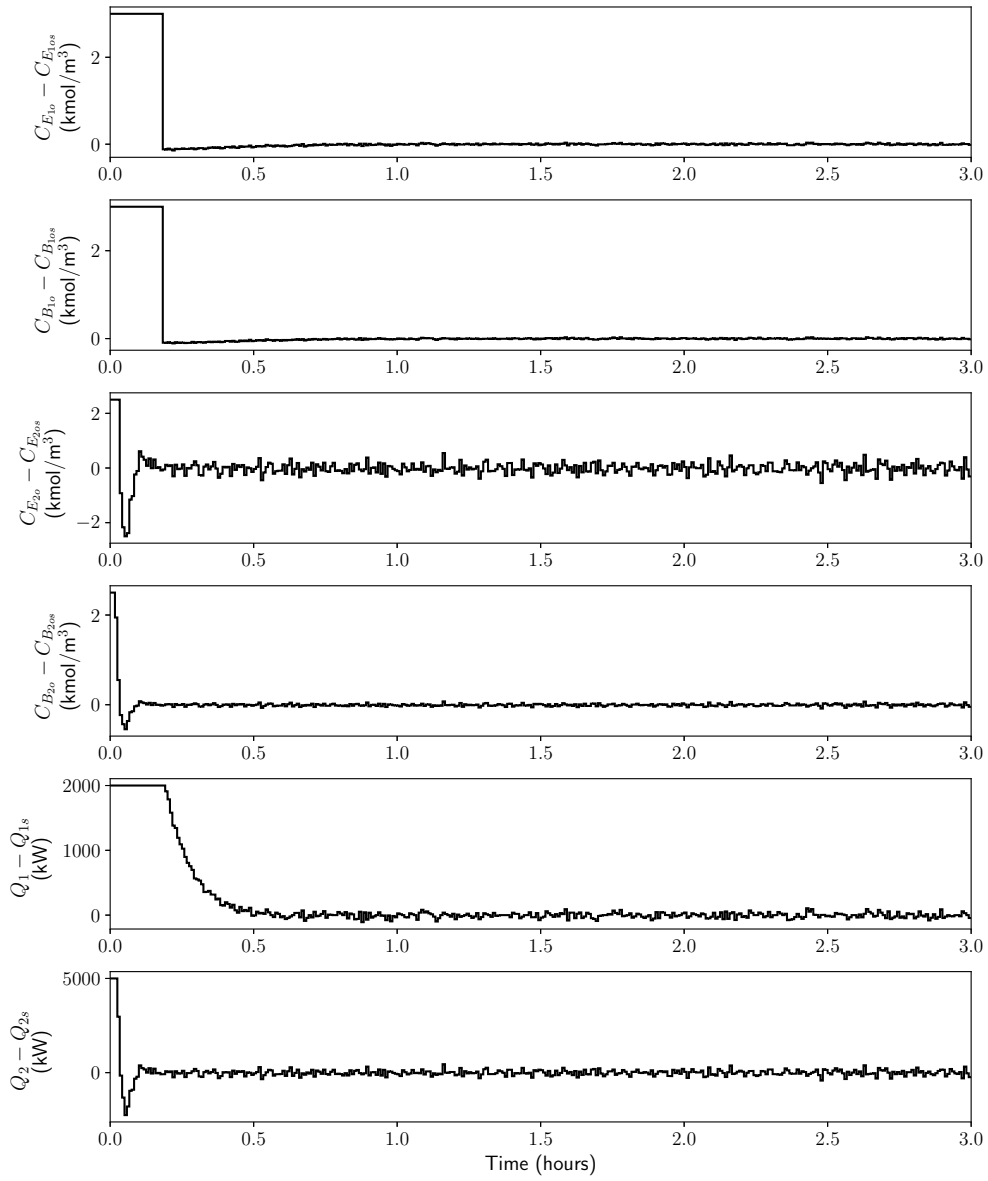

Figure 5.1.6: Control input profiles under the encrypted sequential distributed LMPC framework for the first set of initial conditions.

$\Omega_{\rho_{\min}}$ in approximately 1.5 hours for the first set of initial conditions and 1 hour for the second set.

In the case of the first set of initial conditions, the normalized sum of the control cost function is 1 for the encrypted sequential distributed LMPC and 0.9907 for the encrypted iterative distributed LMPC. For the second set of initial conditions, it is 1 for the encrypted sequential LMPC

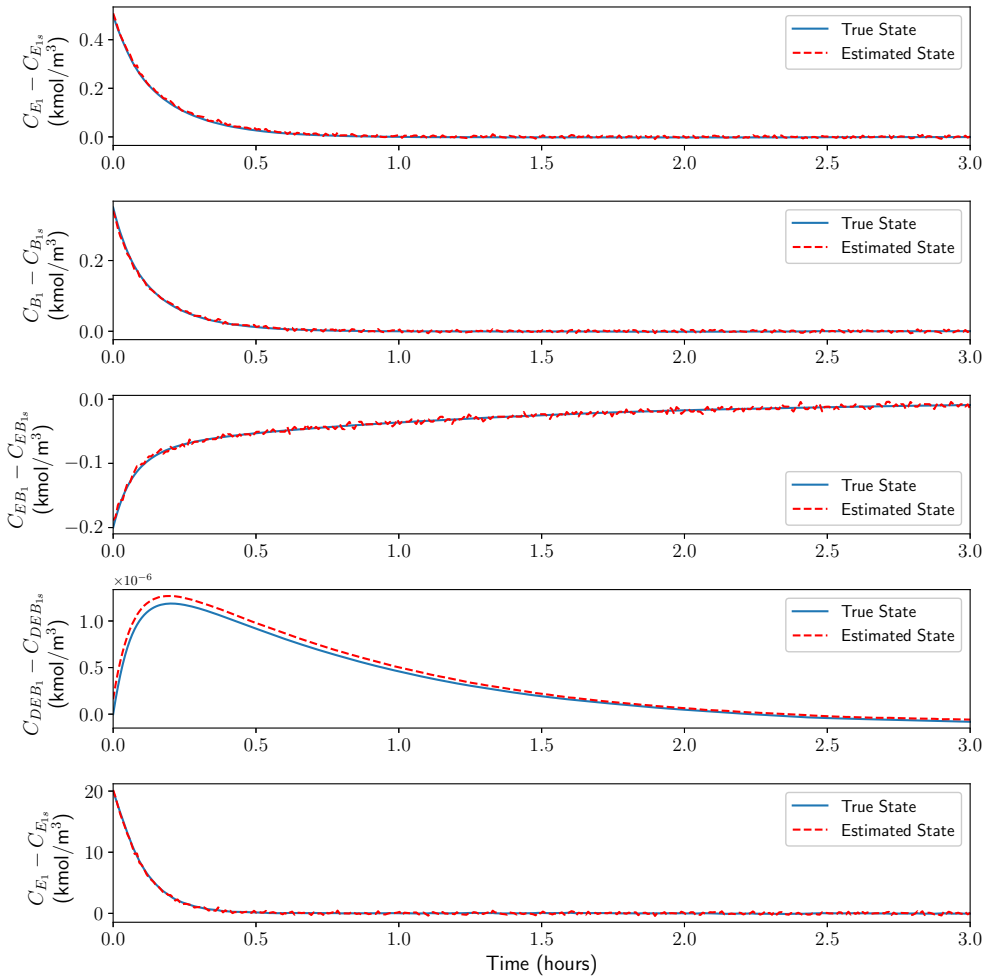

Figure 5.1.7: True state profiles (blue solid line) and estimated state profiles (red dashed line) of CSTR 1 under the encrypted iterative distributed LMPC framework for the second set of initial conditions.

and 0.9884 for the encrypted iterative LMPC. The iterative LMPC outperforms the sequential approach because, in the iterative framework, both LMPCs share and recalculate their control inputs, while, in the sequential framework, LMPC 2 computes control inputs based on an assumption of the stabilizing control law for LMPC 1. The following section provides a detailed comparative analysis of the sequential and iterative DMPCs. Visual results are provided exclusively for the encrypted sequential distributed LMPC demonstrating the performance under the first set of initial conditions and for the encrypted iterative distributed LMPC under the second set of initial

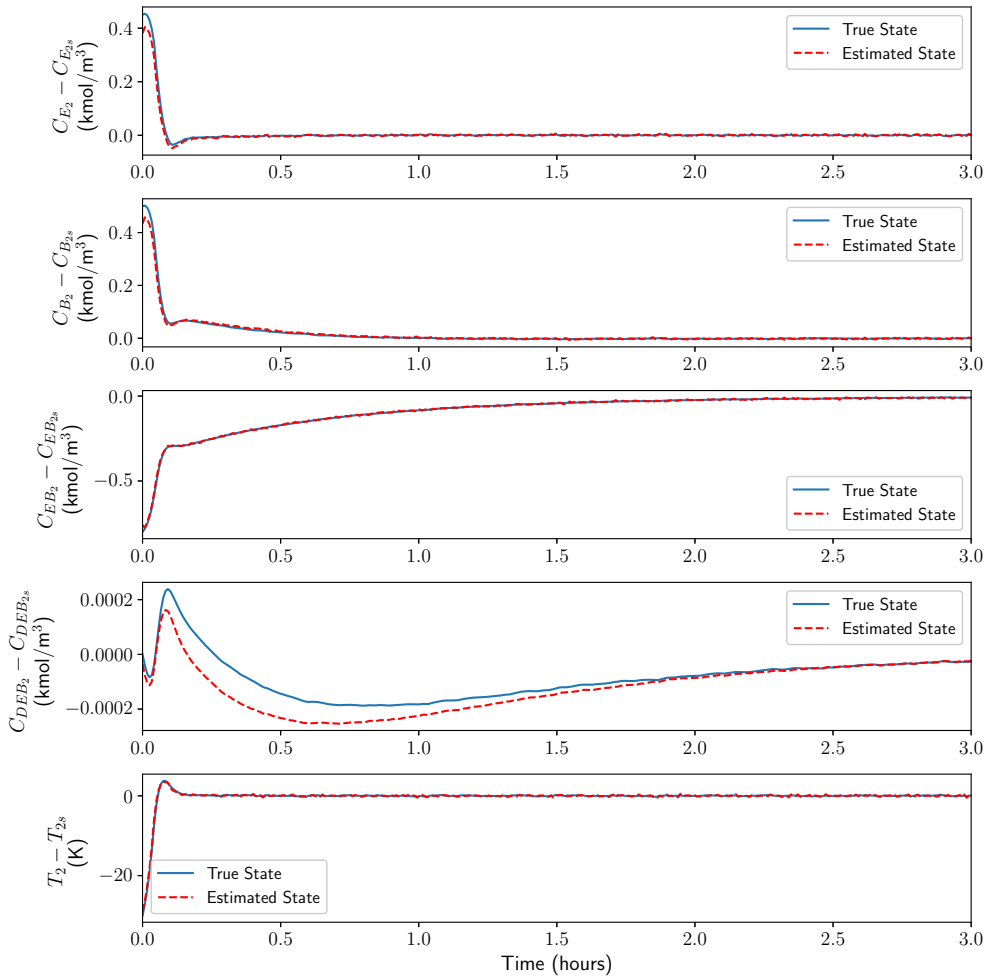

Figure 5.1.8: True state profiles (blue solid line) and estimated state profiles (red dashed line) of CSTR 2 under the encrypted iterative distributed LMPC framework for the second set of initial conditions.

conditions. Notably, when the alternative DMPC system was applied to both initial conditions, the differences in the closed-loop state trajectories were not significant, as evidenced by the close values of the normalized sum of the control cost functions in both scenarios.

Remark 5.1.10. *The encrypted distributed LMPC systems explored in this study involved encrypting and decrypting data as outlined in Section* 5.1.3*, which can lead to errors due to quantization. [81] demonstrated quantization effects in the context of a first-principles-based LMPC and pro-*

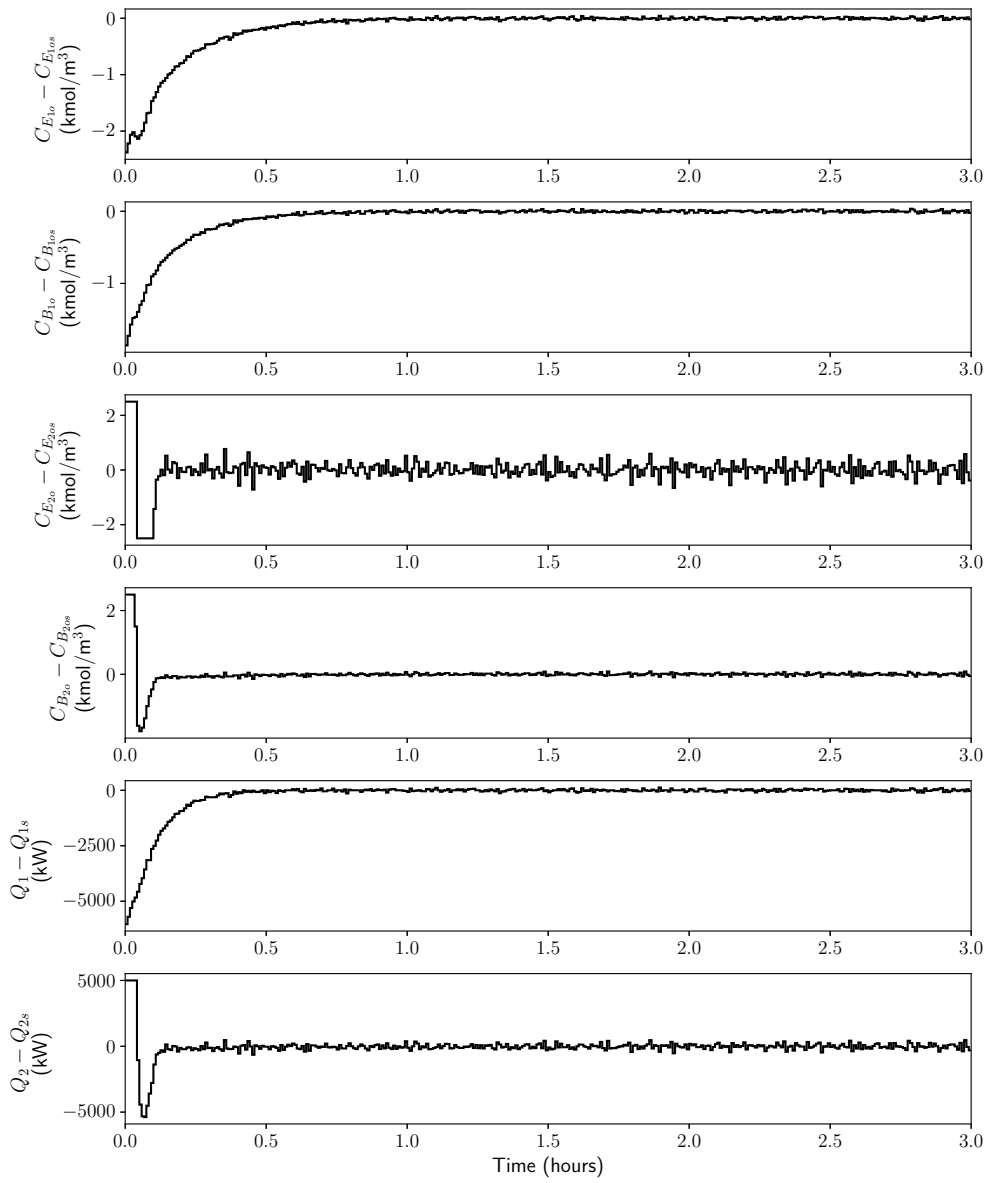

Figure 5.1.9: Control input profiles under the encrypted iterative distributed LMPC framework for the second set of initial conditions.

cess model. Additionally, [39] highlighted the potential for quantization-induced errors to exceed model mismatch errors when different models are employed in the LMPC and in the controlled process. To minimize the quantization error, both works recommended using a higher quantization parameter d. With $d = 8$, both works reported almost identical closed-loop results with encryption

compared to without encryption. Thus, we have used the quantization parameter $d = 8$ for all simulations in this work.

Remark 5.1.11. *Although the LMPC and state estimation models used in this work are first-principles-based, data-based models employing artificial neural networks can also be used in the predictor and LMPC. [3] used machine-learning-based models for the ELO model and LMPC while simulating a first-principles-based process with partial state feedback, showcasing the effectiveness of the state estimator in the presence of plant/model mismatch.*

5.1.5 Comparative analysis of encrypted centralized, decentralized, and distributed LMPC architectures

In this section, we provide a concise overview of the encrypted centralized and decentralized control architectures, both of which incorporate state estimation. Following this, we offer an in-depth comparative analysis that covers the encrypted centralized, decentralized, and distributed LMPCs.

5.1.5.1 Encrypted centralized MPC with state estimation

In Figure 5.1.10, the diagram illustrates the flow of information within an encrypted centralized LMPC system that incorporates state estimation. At time $t = t_k$, where k signifies the sampling instance, the sensors encrypt the measurements denoted as $y(t_k)$ and transmit the resulting ciphertext c to the computing unit responsible for computing all the control inputs. Upon arrival, the data is decrypted, and the quantized states $\hat{y}(t_k)$ are utilized by the state estimator to estimate all states of the system $\bar{x}(t_k)$. These estimated states initialize the LMPC model, enabling it to compute the

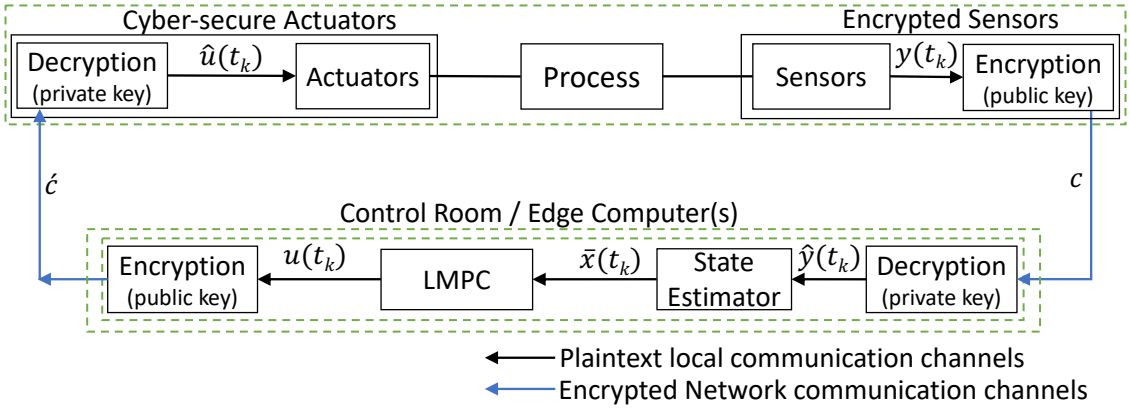

Figure 5.1.10: Illustration of the encrypted centralized control structure.

control inputs, $u(t_k)$. Subsequently, these control inputs are encrypted into the ciphertext $\acute{c}$ and transmitted to the actuators, where it is decrypted to the quantized control inputs $\hat{u}(t_k)$ and applied to the process. Thus, in this approach, only a single computing unit that receives and transmits encrypted signals is utilized for all computations. Additional details and formulation of the LMPC equations of the centralized LMPC can be obtained in [39].

5.1.5.2 Encrypted decentralized MPC with state estimation

In Figure 5.1.11, the diagram illustrates the flow of information within an encrypted decentralized LMPC system with state estimation. Here, the overall system is divided into multiple subsystems, with each subsystem independently computing its control inputs in separate computing units. There is no information exchange of the control inputs between subsystems. At time $t = t_k$, where k is the sampling instance, the sensors encrypt the measurements represented as $y(t_k)$ and transmit the resulting ciphertext c to all the computing units responsible for calculating the control inputs of different subsystems. The ciphertext c is decrypted in each computing unit, and the quantized states $\hat{y}(t_k)$ are used by the state estimator in each subsystem to estimate all states of the entire

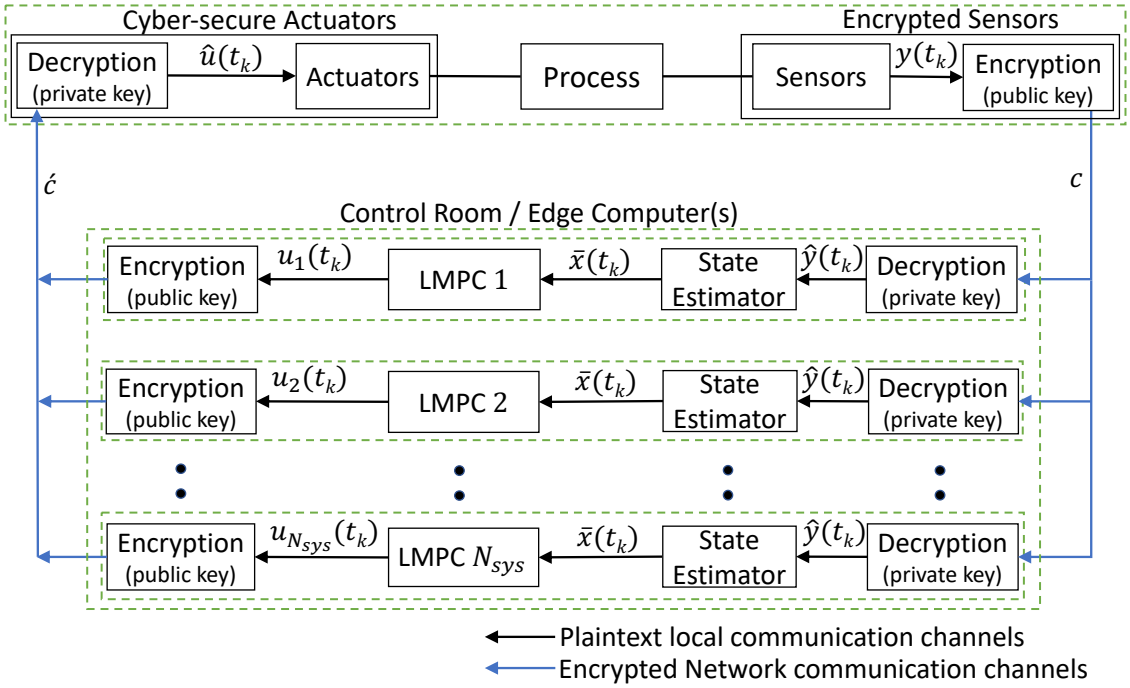

Figure 5.1.11: Illustration of the encrypted decentralized control structure.

system $\bar{x}(t_k)$. All the state estimators shown in Figure 5.1.11 are identical, as each LMPC in the decentralized control framework receives full state feedback to compute the control inputs of its respective subsystem. The LMPC model in each subsystem is then initialized by these estimated states, and is used to compute the control inputs of its respective subsystem only, $u_j(t_k)$. Here j represents the j^{th} subsystem. Subsequently, all control inputs are encrypted and transmitted to their respective actuators, where the ciphertext $\acute{c}$ is decrypted to the quantized control inputs $\hat{u}(t_k)$ and applied to the process. Thus, multiple computing units (equal to the number of subsystems) that receive and transmit encrypted signals are utilized for all computations, which are carried out in an independent and isolated manner. Additional details and formulation of the LMPC equations of the decentralized LMPC can be obtained in [14].

5.1.5.3 Comparison of the encrypted centralized, decentralized, and distributed LMPCs with state estimation

In our analysis, we applied the same system as the example described in Section 5.1.4, using the second set of initial conditions mentioned in the preceding section. Our objective was to compare the computation time and performance of various control architectures in computing the control inputs. Table 5.1.1 provides a summary of the total computation time required for computing control inputs and the normalized sum of the control cost functions for the encrypted centralized, decentralized, and distributed LMPCs.

Table 5.1.1: Computational time and performance of the encrypted centralized, decentralized, sequential distributed, and iterative distributed LMPCs

Control architecture	Average control input computation time	Normalized sum of the control cost function
Centralized MPC	$15.28\,\mathrm{s}$	1
Decentralized MPC	$2.87\,\mathrm{s}$	0.9751
Sequential DMPC	$4.03\,\mathrm{s}$	0.9817
Iterative DMPC	$5.22\,\mathrm{s}$	0.9703

To determine the computation time of the centralized framework, we calculated the time spent by the LMPC in computing the control inputs for the system at each sampling instance. In the case of the decentralized framework, we recorded the longer of the two LMPC computation times at each sampling instance. For the sequential distributed LMPC, we summed the time spent by both LMPCs at each sampling instance to obtain the total control input computation time for that specific sampling instance. In case of the iterative distributed LMPC, we recorded the higher value of the time spent by the two LMPCs at each iteration, and summed these values for the two iterations to calculate the total time spent for computing control inputs, at a particular sampling

instance. It was ensured that the control input computation time at any interval was not only smaller than the sampling period of 30 seconds but also satisfied Eq. (5.1.21). Figure 5.1.12 displays the computation times for all 4 cases at each sampling instance of process operation. Based on the results from Table 5.1.1 and Figure 5.1.12, we can conclude that the decentralized LMPC required the shortest computational time, while the iterative distributed LMPC exhibited the best performance. In contrast, the centralized LMPC not only had the longest computational time but also demonstrated inferior performance compared to the distributed and decentralized LMPC systems.

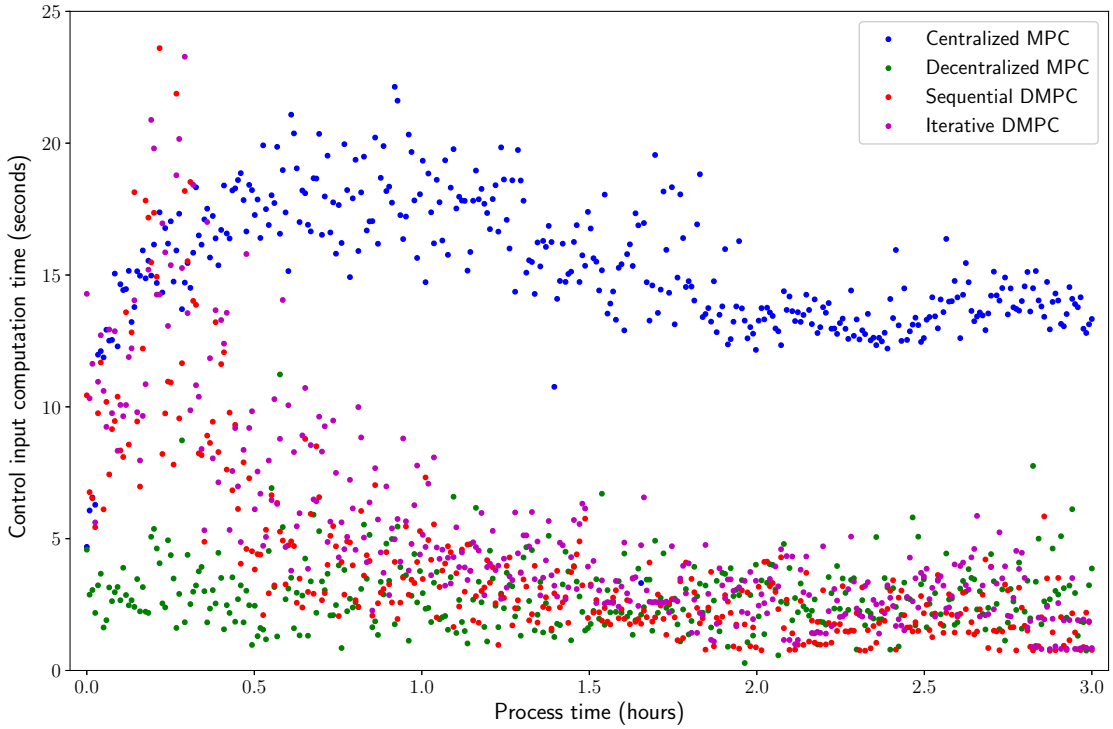

Figure 5.1.12: Control input computation time for the encrypted centralized, decentralized, sequential distributed, and iterative distributed LMPCs at every sampling instance.

The reason behind the slightly improved performance observed with the decentralized LMPC can be attributed to the sequential flow sheet of the process network, featuring two CSTRs in se-

ries. This characteristic renders decentralized LMPC a more suitable and well-conditioned choice compared to centralized LMPC when addressing the optimization problem. Furthermore, the iterative distributed LMPC, which shares control input information with other subsystems during each iteration, demonstrates superior performance compared to both the decentralized LMPC and the sequential distributed LMPC. It is essential to note that this performance enhancement may not be universally applicable to all nonlinear systems, but the enhanced computational efficiency of decentralized and distributed frameworks over centralized ones can indeed be extended to other large-scale systems.

In general, the advantages and disadvantages of all 4 control schemes can be summarized in the following manner:

1. **Centralized MPC:** It offers the advantage of requiring only a single computing unit for all computations, simplifying information flow and reducing costs, which makes it suitable for small systems. All signals transmitted to and received from the remote edge computing unit can be encrypted. However, it comes with a significantly higher computation time compared to the decentralized and distributed MPCs, making it less viable for large processes with numerous states and control inputs. It is, nevertheless, a suitable choice for small-scale systems, where only a single computing unit is needed.

2. **Decentralized MPC:** This approach stands out with the shortest computation time among the four systems, and can even outperform the closed-loop performance of the centralized MPC in specific cases. Furthermore, a decentralized MPC framework does not require communication between different computing units, making it particularly well-suited for large

systems partitioned into many subsystems, where the coupling between subsystems is not very significant. Also, all signals transmitted to and received from the different computing units remain encrypted. However, its performance may deteriorate in cases where the overall system is partitioned into highly coupled subsystems.

3. **Sequential DMPC:** While a decentralized MPC can perform better than the centralized MPC in some cases, it calculates control inputs independently, without any information exchange among subsystems. The integration of information exchange between subsystems can be achieved through the use of a sequential DMPC. In the example discussed, the overall system was only partitioned into two subsystems, and the control inputs were computed in series for the two subsystems. However, this approach may be slower for very large systems partitioned into numerous subsystems, not making it a viable option in that case. The major advantage of a sequential DMPC over an iterative DMPC lies in its reduced communication among subsystems, as information flows only in one direction (from higher to lower subsystems). It is suitable for cases where communication between controllers is necessary, implementing iterative DMPC is not feasible, and the number of subsystems is not extensive. Further, it would be more suitable when Ethernet crossover communication cannot be established between different computing units (if they are placed in different locations) because the communication load of a sequential DMPC is much less compared to an iterative DMPC, and, hence, encrypted signals could be used for internal communication between subsystems, as long as the overall system is not partitioned into numerous subsystems.

4. **Iterative DMPC:** This approach delivers the best overall performance, although it entails

longer computation times compared to decentralized and sequential DMPCs. In the example presented, we considered only two MPCs in the partition, but for systems with more partitions, it can outperform the sequential DMPC in terms of computation time. However, implementing this system requires multiple computing units compared to a single unit in a centralized MPC. Moreover, these units must be located in the same room to establish secure Ethernet crossover communication between them. On the other hand, a decentralized MPC allows for computing units to be located in different locations. Also, the communication load between subsystems in an iterative DMPC is higher than a sequential DMPC, as control input trajectories are shared with all other subsystems multiple times within a single sampling instance. Therefore, it is most suitable for very large systems partitioned into numerous subsystems, especially when these subsystems exhibit a substantial coupling effect with one another, and in situations where secure internal communication channels between different subsystems can be easily established.

To summarize, this section offered a general overview of the advantages, disadvantages, and the suitability of various control architectures with encryption. The decision on which control framework to practically implement should be based on several factors, such as the specific characteristics (size, coupling effect, etc.) of the system to be controlled, the available resources, the desired level of control performance, the budget allocated for computing hardware, and other pertinent considerations.

5.1.6 Conclusion

In this chapter, we introduced and applied encrypted distributed control architectures, both sequential and iterative, with state estimation, to a large-scale nonlinear chemical process network utilizing partial state feedback with sensor noise. We established practical guidelines for implementing this control structure in any nonlinear process by including the selection of key parameters such as l_1, l_2, and d for quantization, and the criterion for setting the sampling time. Through closed-loop simulations, we demonstrated that both the sequential and iterative distributed LMPCs, with encrypted communication between the sensor–controller and controller–actuator links, could stabilize the system within the desired stability region using the extended Luenberger observer for state estimation, in a finite process simulation time. Furthermore, we conducted a comprehensive comparative analysis of various encrypted control strategies, including centralized, decentralized, and distributed approaches with state estimation. The computational time, closed-loop performance, and suitability of the different encrypted control architectures were discussed. In conclusion, our findings indicate that the encrypted iterative distributed LMPC emerges as the most suitable choice for enhancing the cybersecurity of large and complex systems, with highly coupled dynamics between states. This approach reduces the computational complexity associated with centralized control, leverages controller communication to improve closed-loop performance, and maintains a reasonable computation time, while enhancing the cybersecurity of the control system.

Chapter 5.2

Encrypted distributed model predictive control of nonlinear processes

5.2.1 Introduction

In recent years, networks have emerged as pivotal components within manufacturing systems, replacing traditional point-to-point communications across various levels. At the field level, networks have elevated connectivity among sensors, actuators, and controllers, enabling efficient data transfer within the factory floor, while concurrently reducing wiring and minimizing potential points of failure. At the supervisory and management level, networks have facilitated automated plant-wide communication via SCADA (Supervisory Control and Data Acquisition) systems. This has, in turn, expanded data storage capacities and visibility, enabling operational trend analysis and improved decision-making for enhanced closed-loop performance, and has augmented interconnectivity of various parts of the plant.

However, these advantages come with a substantial reliance on networked communications,

whether through the internet or wireless local area networks (LAN), which could be vulnerable to cyber threats. Any compromise within these systems could lead to significant consequences, such as critical service disruption, physical harm, financial loss, and potential threats to public safety. Real-world cyberattack instances underscore the need of cybersecurity measures in networked cyber-physical systems. For instance, the 2015 cyberattack on Ukraine's power grid managed by SCADA controls, led to widespread power outages [45]. Similarly, in 2021, hackers launched a DarkSide ransomware attack on Colonial Pipeline by encrypting networked communication and demanded a ransom for decryption keys. This incident forced Colonial Pipeline to halt operations, leading to extensive disruptions in fuel distribution [82].

PID (Proportional-Integral-Derivative) controllers and PLCs (Programmable Logic Controllers) have been extensively used and continue to be utilized for controlling system states in a decentralized manner. Their decentralized operation reduces computational burden and interdependencies between different controllers. However, in systems with highly coupled process states, where the control inputs applied by one controller directly impacts the controlled states of another controller, traditional controllers might not yield adequate closed-loop performance. To overcome this constraint, complex processes have been effectively managed using model predictive controllers (MPCs). MPCs utilize a mathematical model of the process, obtained from either first-principles or data, to predict future closed-loop state evolution within a defined horizon. Subsequently, control inputs are optimized based on real-time sensor feedback, considering interactions between all states and inputs. This methodology enhances control precision while minimizing utility costs.

For systems regulated by MPCs, at each sampling instance, a nonlinear optimization problem has to be solved to compute optimal control input trajectories, which can be very complex for

large-scale systems involving numerous states and control inputs. To cope with this, distributed MPCs have been proposed [53]. Networked communication has facilitated distributed control systems to be easily established within a SCADA control architecture by enhancing connectivity and data transfer between different controllers without needing elaborate wired communication. However, as mentioned earlier, this has also made the system more vulnerable to cyberattacks with the evolution of technology. Considerable research efforts have been dedicated to areas such as employing linear encrypted controllers [18, 20], developing machine learning-based cyberattack detectors [2, 26], utilizing encrypted decentralized MPCs [37], and creating cyberattack-resilient controllers [68].

Addressing the aforementioned challenges, this work focuses on an encrypted iterative distributed MPC comprising a set of Lyapunov-based MPCs, utilizing encrypted networked communication for communication between sensors, actuators, and computing units responsible for calculating the control inputs. Following the formulation of the proposed control system, a thorough stability analysis is conducted to establish bounds, ensuring system stabilization within the desired stability region. Closed-loop simulations of the encrypted distributed LMPC system implemented in a nonlinear chemical process network are presented and compared with an encrypted centralized LMPC.

5.2.2 Preliminaries

5.2.2.1 Notation

The symbol $\|\cdot\|$ denotes the Euclidean norm of a vector, and $x^\top$ represents the transpose of a vector x. The sets of real numbers, integers, and natural numbers are denoted by $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$, respectively. The additive groups of integers modulo M are represented by $\mathbb{Z}_M$. The symbol "\" denotes set subtraction, where $A \backslash B$ represents the set of elements in set A but not within set B. A function, $f(\cdot)$, is classified as $\mathcal{C}^1$ when it is continuously differentiable in a defined domain. The least common multiple of the integers i and j is denoted by $\mathrm{lcm}(i, j)$. The greatest common divisor that divides i and j with no remainder is denoted by $\gcd(i, j)$.

5.2.2.2 Class of systems

In this research, we consider a general category of nonlinear systems regulated by multiple unique sets of control inputs. Each distinct set of control inputs manages a particular subsystem of the process. To simplify notations, we examine two subsystems—subsystem 1 and subsystem 2—each governed solely by u_1 and u_2, respectively. However, this same analysis can be extended to any nonlinear system controlled by N_{sys} subsystems regulated by N_{sys} unique sets of manipulated inputs. While partitioning a large-scale nonlinear process, manipulated inputs that have a strong, direct effect on certain states should be grouped together and be manipulated by the same controller. The work of [71] describes such methods in detail, and can be an area of future research. The overall nonlinear system is characterized by a set of ordinary differential equations (ODEs),

formulated in the following manner:

$$\begin{aligned} \dot{x} &= f(x(t), u_1(t), u_2(t), w(t)) \\ y &= x + v \end{aligned} \tag{5.2.1}$$

The state vector is denoted by $x \in \mathbb{R}^n$, while $y \in \mathbb{R}^n$ is the vector of state measurements that are sampled continuously. $u_1 \in \mathbb{R}^{m_1}$ and $u_2 \in \mathbb{R}^{m_2}$ represent the sets of control inputs, $w \in \mathbb{R}^w$ is the disturbance vector, and $v \in \mathbb{R}^n$ is the noise vector. The control input constraints are defined by $u_1 \in U_1 := \{u_{1_i}^{\min} \leq u_{1_i} \leq u_{1_i}^{\max}, i = 1, \ldots, m_1\}, \subset \mathbb{R}^{m_1}$, and $u_2 \in U_2 := \{u_{2_i}^{\min} \leq u_{2_i} \leq u_{2_i}^{\max}, i = 1, \ldots, m_2\}, \subset \mathbb{R}^{m_2}$. $u = [u_1\ u_2]^\top \in U$ is the bounded control input vector formed by concatenating u_1 and u_2. The vector function $f(\cdot)$ is locally Lipschitz with respect to its arguments. We consider $f(0, 0, 0, 0) = 0$, such that the steady state of Eq. (5.2.1) is the origin. Without loss of generality, we set the initial time to zero ($t_0 = 0$). $S(\Delta)$ is defined as a collection of piece-wise constant functions characterized by an interval of Δ.

5.2.2.3 Stability assumptions

Accounting for interactions between the partitioned subsystems of the nonlinear process, we assume the existence of stabilizing control laws $u_1 = \Phi_1(x) \in U_1$, $u_2 = \Phi_2(x) \in U_2$ which regulate subsystems 1 and 2, respectively, such that the system of Eq. (5.2.1) with $w \equiv 0$ and $v \equiv 0$ is rendered exponentially stable, signifying the existence of a $\mathcal{C}^1$ control Lyapunov function $V(x)$ that satisfies the subsequent inequalities for all $x \in \mathbb{R}^n$ within D, which is an open region around the origin:

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2, \tag{5.2.2a}$$

$$\frac{\partial V(x)}{\partial x} f(x, \Phi_1(x), \Phi_2(x), 0) \leq -c_3|x|^2, \tag{5.2.2b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq c_4|x|, \tag{5.2.2c}$$

where c_i, $i = \{1, 2, 3, 4\}$ are positive constants. In the nonlinear system of Eq. (5.2.1), the closed-loop stability region can be defined as Ω_ρ, which is a level set of the control Lyapunov function V. In particular, $\Omega_\rho := \{x \in D | V(x) \leq \rho\}$, where $\rho > 0$. Thus, starting from any initial condition in Ω_ρ, the control inputs $\Phi_1(x)$ and $\Phi_2(x)$ guarantee that the state trajectory of the closed-loop system remains inside Ω_ρ. Further, based on the Lipschitz property of $f(x, u_1, u_2, w)$ and the bounds on u_1, u_2, and w, the subsequent inequalities hold for all $x \in \Omega_\rho, u_1 \in U_1, u_2 \in U_2$ and $w \in W$ with positive constants M_F and L'_w:

$$|f(x, u_1, u_2, w)| \leq M_F, \tag{5.2.3a}$$

$$\left| \frac{\partial V}{\partial x} f(x, u_1, u_2, w) - \frac{\partial V}{\partial x} f(x, u_1, u_2, 0) \right| \leq L'_w|w|. \tag{5.2.3b}$$

5.2.2.4 Paillier cryptosystem

In this study, we utilize the Paillier cryptosystem [67] in order to encrypt all signals that are transmitted through the networked communication established. While we do not leverage the semi-homomorphic nature of the additive homomorphism within the Paillier cryptosystem, it is incorporated to enable the integration of conventional controllers, like PI (proportional-integral) controllers, that can calculate control inputs in an encrypted space, within the control architecture if needed [41]. Prior to encryption, we generate the public key (for encryption) and the private key (for decryption) and can be outlined as follows:

1. Choose two large prime integers (p and q) randomly, such that, $\gcd((p-1)(q-1), pq) = 1$.

2. Define, $M = pq$.

3. Select a random integer $\hat{g} \in \mathbb{Z}_{M^2}$, where $\mathbb{Z}_{M^2}$ is the multiplicative group of integers modulo M^2.

4. Compute $\lambda = \text{lcm}(p-1, q-1)$.

5. Specify $\hat{L}(x) = (x-1)/M$.

6. Verify the existence of the subsequent modular multiplicative inverse:
   $u = (\hat{L}(\hat{g}^{\lambda} \bmod M^2))^{-1} \bmod M$.

7. If the inverse does not exist, revisit step 3 and select an alternate value of $\hat{g}$. If the inverse exists, $(M, \hat{g})$ is the public key and (λ, u) is the private key.

Upon obtaining the keys, authorized recipients receive the public key for encryption and the private key for decryption. Encryption is executed in the following manner:

$$E_M(m, r) = c = \hat{g}^m r^M \bmod M^2 \tag{5.2.4}$$

where r is an integer randomly chosen from the set $\mathbb{Z}_M$, and c denotes the resulting ciphertext by encrypting m. Decryption is executed in the following manner:

$$D_M(c) = m = \hat{L}(c^{\lambda} \bmod M^2)u \bmod M \tag{5.2.5}$$

Remark 5.2.1. *Traditional approaches, such as mapping floating points to a set or applying mathematical transformations to achieve data privacy, may prove inadequate in practice. When these methods are used during steady-state operation, it results in the transmission of identical values. On the other hand, during encryption, a distinct random number is generated each time data is encrypted. This feature ensures that encrypting identical numbers results in different ciphertexts, thereby significantly enhancing cybersecurity measures. While encryption enhances privacy, it also enhances cybersecurity by protecting the system against intelligent cyberattacks as discussed in the work of [41].*

5.2.2.5 Quantization

For the utilization of the Paillier cryptosystem, the data intended for encryption is required to be in the form of natural numbers within $\mathbb{Z}_M$. However, prior to encryption, the signal values exist in floating-point format. Thus, we implement quantization, to map the floating-point numbers into $\mathbb{Z}_M$ [19]. Employing a signed fixed-point binary representation, we establish a set, $\mathbb{Q}_{l_1,d}$, characterized by parameters l_1 and d. The parameter l_1 is defined as the total bit count (integer and fractional), and d denotes the fractional bits. The number of fractional bits represents the number of bits used to represent the fractional part of the floating point data. It is equal to the quantization parameter. The $\mathbb{Q}_{l_1,d}$ set encompasses rational numbers ranging from -2^{l_1-d-1} to $2^{l_1-d-1} - 2^{-d}$, with increments of 2^{-d}. For a number q in $\mathbb{Q}_{l_1,d}$, there exists $\beta \in \{0,1\}^{l_1}$, such that $q = -2^{l_1-d-1}\beta l_1 + \sum_{i=1}^{l_1-1} 2^{i-d-1}\beta_i$. The function $g_{l_1,d}$ maps a real-number data point a to the set

$\mathbb{Q}_{l_1,d}$, and is defined by the following equation,

$$\begin{aligned} g_{l_1,d} &: \mathbb{R} \to \mathbb{Q}_{l_1,d} \\ g_{l_1,d}(a) &:= \arg\min_{q \in \mathbb{Q}_{l_1,d}} |a - q| \end{aligned} \tag{5.2.6}$$

Following this, the quantized data undergoes a transformation into a set of positive integers ($\mathbb{Z}_M$) via a bijective mapping ($f_{l_2,d}$), as detailed in [19]:

$$\begin{aligned} f_{l_2,d} &: \mathbb{Q}_{l_1,d} \to \mathbb{Z}_{2^{l_2}} \\ f_{l_2,d}(q) &:= 2^d q \bmod 2^{l_2} \end{aligned} \tag{5.2.7}$$

In the encryption process, integer plaintext messages from the set $Z_{2^{l_2}}$ are transformed into ciphertexts, and can then be decrypted back to set $Z_{2^{l_2}}$. To retrieve the original data point belonging to the set $\mathbb{Q}_{l_1,d}$, we define an inverse mapping denoted as $f^{-1}_{l_2,d}$ in the following manner:

$$f^{-1}_{l_2,d} : \mathbb{Z}_{2^{l_2}} \to \mathbb{Q}_{l_1,d} \tag{5.2.8}$$

$$f^{-1}_{l_2,d}(m) := \frac{1}{2^d} \begin{cases} m - 2^{l_2} & \text{if } m \geq 2^{l_2-1} \\ m & \text{otherwise} \end{cases} \tag{5.2.9}$$

5.2.3 Development of the encrypted iterative distributed LMPC

5.2.3.1 Design of the encrypted iterative distributed LMPC

Figure 5.2.1 illustrates the control structure of the encrypted iterative distributed LMPC, where all LMPCs collaboratively optimize control actions for their respective subsystems. The sampling

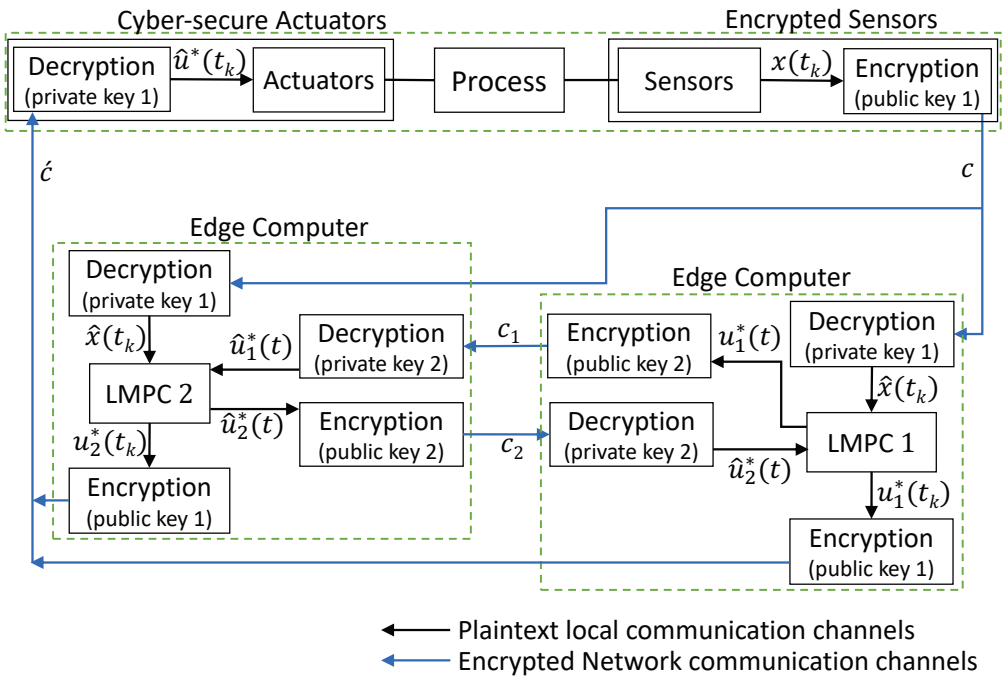

Figure 5.2.1: Block diagram of the encrypted iterative distributed LMPC system.

period represents the time between two consecutive measurements during which a constant control input is maintained by the actuators in a sample-and-hold manner. A total of two LMPCs that utilize the complete process model for computing a set of distinct control inputs has been considered to present the control strategy. A single iteration of an LMPC corresponds to an optimal control input computation by an LMPC, which may be repeated with updated input information from the other LMPC, if the termination criterion is not satisfied. The control strategy can be implemented through the following steps:

1. At time $t = t_k$, where k is the current sampling instance, using public key 1, signals $x(t_k)$ from sensors are encrypted to ciphertext c and transmitted to the computing units of distinct control subsystems.

2. In each unit, using private key 1, the encrypted signals are decrypted. The resulting quantized states $\hat{x}(t_k)$ initialize the LMPC model.

3. At iteration $z = 1$, LMPC 1 computes the optimal control input trajectory $u_1^*(t)$, using the quantized states $\hat{x}(t_k)$, and assuming the stabilizing control law $u_2(t) = \Phi_2(\hat{x}(t))$ for the second subsystem, for $t \in [t_k, t_{k+N})$, where N is the prediction horizon. In parallel, LMPC 2 computes the optimal control input trajectory $u_2^*(t)$ assuming $u_1(t) = \Phi_1(\hat{x}(t))$, the stabilizing controller for the first subsystem, for $t \in [t_k, t_{k+N})$.

4. At the end of the first iteration, LMPC 1 and LMPC 2 encrypt their computed control inputs over the prediction horizon using public key 2, to the ciphertexts c_1 and c_2, respectively. Subsequently, LMPC 1 decrypts c_2 to obtain the quantized control input of LMPC 2, $\hat{u}_2^*(t)$, and LMPC 2 decrypts c_1 to obtain the quantized control input of LMPC 1, $\hat{u}_1^*(t)$, for $t \in [t_k, t_{k+N})$.

5. At iteration $z = 2$, both LMPCs recalculate the optimal control inputs of their subsystem using the quantized control inputs (after decryption) of the other subsystems. Subsequently, the new control input trajectories are again shared with the other LMPCs, as described previously. The aforementioned steps are reiterated till a termination condition is satisfied, which could be the number of iterations, or the difference between computed control inputs in successive iterations is less than a specified threshold value.

6. Upon meeting this termination condition, both LMPCs encrypt their optimal control inputs for the subsequent sampling period (utilizing public key 1) and transmit the ciphertexts to

the corresponding actuators of each of their respective subsystem.

7. At the actuator, the ciphertext $\acute{c}$ undergoes decryption using private key 1 to yield $\hat{u}^*(t_k)$, the quantized input, which is applied to the process.

By encrypting all signals in a distributed setting between the sensors, controllers, and actuators, secure information exchange is established between computing units situated at various locations, eliminating the necessity of a control room.

Remark 5.2.2. *In the proposed design, sensor–controller and controller–actuator communication links utilize distinct keys for encryption–decryption compared to the inter-controller communication link. However, a single pair of keys may also serve this purpose. The decision to choose a distinct set of keys aims to meet the specific cybersecurity requirements based on the cyber-physical needs of the system. For instance, in transmitting encrypted signals across the entire plant, higher bit length keys would be recommended. Conversely, when exchanging encrypted signals between various controllers or computing units, lower bit length keys might be sufficient.*

Remark 5.2.3. *The time and computational load required for encrypting signals increases with longer key bit lengths. A 2048-bit key results in an approximately 4096-bit ciphertext and requires about 0.066 seconds for encryption. On the other hand, a 1024-bit key produces a ciphertext of around 2048 bits within 0.0096 seconds. The most recent NIST recommendations suggest using asymmetric keys of 2048 bits, an upgrade from the previous recommendation of 1024 bits [8]. The determination of key lengths should be guided by factors like cyber-physical vulnerability, desired cybersecurity level, and available computational resources. The time estimates were derived from*

encryption processes on an Intel i7-10700K 3.80 GHz computer with 64 GB of RAM. The computational complexity of encryption–decryption varies by $\mathcal{O}(\bar{k}^3)$ as mentioned in the work of [17], where $\bar{k}$ represents the number of bits of the keys utilized. Thus, increasing the bits of the keys significantly increases the computational load for encryption–decryption.

Remark 5.2.4. *In an industrial setting, the standard approach for encryption would involve employing microcontrollers within sensors and actuators to encrypt and decrypt signals, respectively. Encrypted signals are sent to the controllers via RF (radio-frequency) transmission modules. Similarly, actuators receive signals from the RF receiver module. To decrease the total computation time for encryption-decryption, large-scale systems can equip individual sensors and actuators with dedicated microcontrollers and RF modules. This setup enables parallel operations for the transmission and reception of encrypted signals.*

Remark 5.2.5. *A ciphertext encrypted using a 2048-bit key will be roughly 4096 bits or 512 bytes (1 byte = 8 bits). Wireless communication standards like Wi-Fi 4, Wi-Fi 5, and Wi-Fi 6 offer bandwidths ranging from hundreds of megabytes per second (MBps) to a few gigabytes per second. This bandwidth is more than adequate to transmit multiple encrypted ciphertexts at each sampling instance. For instance, a 4096-bit ciphertext would require approximately 1 microsecond for transmission through Wi-Fi with a bandwidth of 500 MBps. Therefore, the transmission of encrypted signals would not significantly burden established communication channels, while reinforcing the cybersecurity of the control system.*

Remark 5.2.6. *To deal with input delays, a state-predictor can be integrated. The state predictor would estimate the state values after the period corresponding to the input delay and the LMPC*

model would be initialized with these predicted states. This has been demonstrated in the work of [37] using an encrypted decentralized LMPC. As the LMPC is initialized with the new predicted states, the same concept can be extended to the encrypted distributed LMPC presented in this research.

5.2.3.2 Quantization errors in the control architecture

The closed-loop configuration presented in Figure 5.2.1 introduces two error sources: one originating from state quantization in the sensor-controller link, while another stemming from control input quantization within the controller-actuator link, which are bounded as follows:

$$|x(t_k) - \hat{x}(t_k)| \leq 2^{-d-1} \tag{5.2.10a}$$

$$|u(t_k) - \hat{u}(t_k)| \leq 2^{-d-1} \tag{5.2.10b}$$

The upper bounds for the quantization error in Eq. (5.2.10) has been derived in [39]. Leveraging the local Lipschitz property, the error for the stabilizing controller of the j^{th} subsystem will be bounded by the following equation, where L'_j is a positive constant, for $x \in \Omega_\rho$, the stability region:

$$|\Phi_j(\hat{x}) - \Phi_j(x)| \leq L'_j|\hat{x} - x| \leq L'_j 2^{-d-1} \tag{5.2.11}$$

5.2.3.3 Encrypted iterative distributed LMPC system

The optimization task for the j^{th} LMPC in the iterative distributed LMPC, during the initial iteration ($z = 1$), is formulated as:

$$\mathcal{J} = \min_{u_j \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), \Phi_m(\tilde{x}(t)), u_j(t))\, \mathrm{d}t,$$

$$\text{where } m = 1, 2 \text{ and } m \neq j \tag{5.2.12a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t), \Phi_m(\tilde{x}(t)), u_j(t)) \tag{5.2.12b}$$

$$u_j(t) \in U_j,\ \forall\, t \in [t_k, t_{k+N}) \tag{5.2.12c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{5.2.12d}$$

$$\dot{V}(\hat{x}(t_k), \Phi_m(\hat{x}(t_k)), u_j(t_k)) \leq$$

$$\dot{V}(\hat{x}(t_k), \Phi_m(\hat{x}(t_k)), \Phi_j(\hat{x}(t_k))),$$

$$\text{if } \hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{5.2.12e}$$

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

$$\text{if } \hat{x}(t_k) \in \Omega_{\rho_{\min}} \tag{5.2.12f}$$

For subsequent iterations $z > 1$, following the exchange of the optimal control inputs $u_m^*(t)$ with all the other LMPCs, the optimization task for the j^{th} LMPC is:

$$\mathcal{J} = \min_{u_j \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), \hat{u}_m(t), u_j(t)) \, \mathrm{d}t,$$

$$\text{where } m = 1, 2 \text{ and } m \neq j \tag{5.2.13a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t), \hat{u}_m(t), u_j(t)) \tag{5.2.13b}$$

$$u_j(t) \in U_j, \ \forall \, t \in [t_k, t_{k+N}) \tag{5.2.13c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{5.2.13d}$$

$$\dot{V}(\hat{x}(t_k), \hat{u}_m(t_k), u_j(t_k)) \leq$$

$$\dot{V}(\hat{x}(t_k), \Phi_m(\hat{x}(t_k)), \Phi_j(\hat{x}(t_k))),$$

$$\text{if } \hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{5.2.13e}$$

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

$$\text{if } \hat{x}(t_k) \in \Omega_{\rho_{\min}} \tag{5.2.13f}$$

The key contrast between Eq. (5.2.12) and Eq. (5.2.13) is that in the former, the j^{th} LMPC computes the optimal control inputs for its respective subsystem by assuming the stabilizing control laws for the remaining subsystems, while in the latter, the LMPC uses quantized control inputs of other LMPCs (after decryption) from the previous iteration, to calculate the optimal inputs for its subsystem. $\tilde{x}$ denotes the state trajectory predicted by the LMPC model. The quantized states, denoted as $\hat{x}$, from Eq. (5.2.12d) and Eq. (5.2.13d), initialize the LMPC model for predicting the state trajectory in accordance with Eq. (5.2.12b) and Eq. (5.2.13b), respectively. This prediction is

used to calculate the integral of the cost functions represented by Eq. (5.2.12a) and Eq. (5.2.13a), respectively, to determine the optimized control inputs, $u_j^*(t)$, throughout the prediction horizon. However, the LMPC transmits only the first control input of the sequence which is applied to the system by the actuator within the interval $t \in [t_k, t_{k+1})$, where this process is repeated at each sampling period. Here, k is the sampling instance, while N denotes the number of sampling periods in the prediction horizon. The constraints of Eq. (5.2.12c) and Eq. (5.2.13c) bound the control inputs, and it remains consistent across all iterations for a particular subsystem. The Lyapunov constraint of Eq. (5.2.12e) and Eq. (5.2.13e) bounds the state $x(t_k)$ at time t_k within the region $\Omega_\rho \setminus \Omega_{\rho_{\min}}$, where $\rho_{\min}$ is a level set of V in proximity to the origin. Eq. (5.2.12f) and Eq. (5.2.13f) ensure that the closed-loop state is bounded within $\Omega_{\rho_{\min}}$ once it enters $\Omega_{\rho_{\min}}$.

Remark 5.2.7. *In the LMPC formulation presented, the LMPCs transmit only the control inputs to be implemented by the actuators over the next sampling period. To address challenges related to delayed and/or asynchronous signals, a control logic can be integrated. In instances where sensor signals are absent, the LMPC transmits the control input calculated for the subsequent sampling period during the preceding instance, ensuring continuous operation. This adaptive strategy can be selectively applied by subsystems experiencing signal reception issues within a distributed system. Moreover, a control logic can be devised to transmit the control inputs after the first iteration, if challenges arise in communicating control inputs with other controllers, switching from a distributed to a decentralized setup. Consequently, the utilization of distributed MPC introduces substantial flexibility to adapt control systems according to diverse conditions and practical requirements, all without necessitating extensive modifications.*

5.2.3.4 Robustness of the encrypted distributed LMPC

In this subsection, we will conduct a comprehensive stability analysis of the nonlinear system of Eq. (5.2.1), considering bounded process disturbances. Initially, we ascertain the closed-loop stability using the encrypted stabilizing controllers $\hat{\Phi}_1(\hat{x})$ and $\hat{\Phi}_2(\hat{x})$, and subsequently, we extend our results to evaluate system stability under the encrypted iterative distributed LMPC defined by Eq. (5.2.12) and Eq. (5.2.13).

Theorem 5.2.1. *We consider the system of Eq.* (5.2.1) *with bounded disturbances* $|w| \leq w_m$*, to examine the closed-loop system stability under the encrypted stabilizing controllers* $\hat{\Phi}_1(\hat{x})$ *and* $\hat{\Phi}_2(\hat{x})$*. The stabilizing controllers* $\Phi_1(x)$ *and* $\Phi_2(x)$*, without encryption, complies with the inequalities stated in Eq.* (5.2.2)*. Also, the initial state* x_0 *is assumed to be within the region* $\Omega_{\hat{\rho}}$ *where* $\hat{\rho} < \rho$*. For a sufficiently large time* $T > 0$*, where* T *is defined as the time taken by* $x(t)$ *to enter* $\Omega_{\rho_{\min}}$*, the positive real numbers* $L'_x, L'_{e_1}, L'_{e_2}, M_F, L'_w$*,* $e_1 = (L_1+1)2^{-d-1}$*, and* $e_2 = (L_2+1)2^{-d-1}$ *can be determined, for which* Δ, w, d*, and* $\epsilon_w > 0$ *exist, such that the subsequent conditions are met:*

$$\begin{aligned} &L'_x M_F \Delta + L'_{e_1}|e_1| + L'_{e_2}|e_2| + L'_w|w| - \frac{c_3}{c_2}\rho_s \leq -\epsilon_w \\ &\rho_{\min} = \max\{V(x(t+\Delta))|V(x(t)) \leq \rho_s\} \end{aligned} \tag{5.2.14}$$

where $\rho > \hat{\rho} > \rho_{\min} > \rho_s$*. Then,* $x(t)$*, under the encrypted stabilizing controller, is within* $\Omega_{\hat{\rho}}$ *and ultimately converges to* $\Omega_{\rho_{\min}}$ *for* $t \geq T$*.*

Proof. The time-derivative of the control Lyapunov function for the nonlinear system (Eq. (5.2.1))

with bounded disturbances under the stabilizing control law is:

$$\begin{aligned}\dot{V} &= \frac{\partial V(x(t))}{\partial x} f(x(t), \hat{\Phi}_1(\hat{x}(t_k)), \hat{\Phi}_2(\hat{x}(t_k)), w) \\ &= \frac{\partial V(x(t))}{\partial x} f(x(t), \hat{\Phi}_1(\hat{x}(t_k)), \hat{\Phi}_2(\hat{x}(t_k)), w) \\ &\quad - \frac{\partial V(x(t))}{\partial x} f(x(t), \hat{\Phi}_1(\hat{x}(t_k)), \hat{\Phi}_2(\hat{x}(t_k)), 0) \\ &\quad + \frac{\partial V(x(t))}{\partial x} f(x(t), \hat{\Phi}_1(\hat{x}(t_k)), \hat{\Phi}_2(\hat{x}(t_k)), 0).\end{aligned} \tag{5.2.15}$$

Based on the Lipschitz condition in Eq. (5.2.2) and Eq. (5.2.3b), the subsequent inequality holds:

$$\dot{V} \leq \frac{\partial V(x(t))}{\partial x} f(x(t), \hat{\Phi}_1(\hat{x}(t_k)), \hat{\Phi}_2(\hat{x}(t_k)), 0) + L'_w|w| \tag{5.2.16}$$

Substituting the error bounds resulting due to quantization, as derived in Eq. (5.2.10),

$$\begin{aligned}\dot{V} \leq \frac{\partial V(x(t))}{\partial x} & f(x(t), \Phi_1(\hat{x}(t_k)) \\ & + 2^{-d-1}, \Phi_2(\hat{x}(t_k)) + 2^{-d-1}, 0) + L'_w|w|\end{aligned} \tag{5.2.17}$$

Further, $\Phi_j(\hat{x}(t_k)) = \Phi_j(\hat{x}(t_k)) - \Phi_j(x(t_k)) + \Phi_j(x(t_k))$ for $j = \{1, 2\}$. Using the Lipschitz property, $\Phi_j(\hat{x}(t_k)) - \Phi_j(x(t_k)) \leq L_j|\hat{x} - x| \leq L_j 2^{-d-1}$. When we substitute this in Eq. (5.2.17), we get:

$$\begin{aligned}\dot{V} \leq \frac{\partial V(x(t))}{\partial x} & f(x(t), \Phi_1(x(t_k)) + e_1, \Phi_2(x(t_k)) + e_2, 0) \\ & + L'_w|w|\end{aligned} \tag{5.2.18}$$

where $e_1 = (L_1 + 1)2^{-d-1}$ and $e_2 = (L_2 + 1)2^{-d-1}$ represent the error bounds from quantization.

From the constraints stated in Eq. (5.2.2), we can re-write Eq. (5.2.18) as:

$$\begin{aligned}\dot{V} \leq & \frac{\partial V(x(t))}{\partial x} f(x(t), \Phi_1(x(t_k)) + e_1, \Phi_2(x(t_k)) + e_2, 0) \\ & - \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), \Phi_1(x(t_k)), \Phi_2(x(t_k)), 0) \\ & + \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), \Phi_1(x(t_k)), \Phi_2(x(t_k)), 0) \\ & + L'_w|w|\end{aligned} \tag{5.2.19}$$

From Eq. (5.2.19), we can define $g(x, e_1, e_2) = f(x, \Phi_1(x) + e_1, \Phi_2(x) + e_2, 0)$. In addition, the positive constants, L'_x, L'_ξ, and L'_q exist, such that the subsequent Lipschitz inequality holds for all $x, x' \in \Omega_{\hat{\rho}}$:

$$\begin{aligned}&\left| \frac{\partial V(x)}{\partial x} g(x, e_1, e_2) - \frac{\partial V(x')}{\partial x} g(x', 0, 0) \right| \leq \\ & L'_x|x - x'| + L'_{e_1}|e_1| + L'_{e_2}|e_2|\end{aligned} \tag{5.2.20}$$

Hence, we can re-write Eq. (5.2.19) as:

$$\begin{aligned}\dot{V} \leq & \frac{\partial V(x(t))}{\partial x} g(x(t), e_1, e_2) - \frac{\partial V(x(t_k))}{\partial x} g(x(t_k), 0, 0) \\ & - c_3|x(t_k)|^2 + L'_w|w| \\ \leq & \ L'_x|x(t) - x(t_k)| + L'_{e_1}|e_1| + L'_{e_2}|e_2| \\ & - c_3|x(t_k)|^2 + L'_w|w|\end{aligned} \tag{5.2.21}$$

From the continuity property of $x(t)\ \forall\, t \in [t_k, t_k + \Delta)$, we have $|x(t) - x(t_k)| \leq M_F\Delta, \forall t \in [t_k, t_k + \Delta)$. Utilizing this bound and from the inequalities of Eq. (5.2.2), we can re-write Eq. (5.2.21) as follows:

$$\dot{V} \leq L'_x M_F \Delta + L'_{e_1}|e_1| + L'_{e_2}|e_2| + L'_w|w| - \frac{c_3}{c_2}\rho_s \tag{5.2.22}$$

In Eq. (5.2.22), the first term signifies the error stemming from the sample-and-hold control input implementation, the second and third terms denote quantization errors due to encryption, and the fourth term indicates the error from process disturbances. The aforementioned errors are constrained and can be effectively minimized by utilizing a lower sampling time and a higher quantization parameter for encryption. As a result, the combined sum of these is also constrained and can be rendered suitably small. Hence, there exist positive real numbers Δ, d, and ϵ_w, such that the following inequality holds for all $t \in [0, T]$:

$$L'_x M_F \Delta + L'_{e_1}|e_1| + L'_{e_2}|e_2| + L'_w|w| - \frac{c_3}{c_2}\rho_s \leq -\epsilon_w$$

implying that $\dot{V} \leq -\epsilon_w$ for any $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_s}$ for all $t_k \in [0, T]$. Thus, upon satisfying the conditions of Eq. (5.2.14), under the encrypted stabilizing controller, the closed-loop system state is confined in $\Omega_{\hat{\rho}}$ and converges within $\Omega_{\rho_s} \subseteq \Omega_{\rho_{\min}}$ in time T, and stays within the desired stability region. □

Now, we advance to the stability analysis of the closed-loop system employing the encrypted distributed LMPC.

Theorem 5.2.2. *We consider the system of Eq.* (5.2.1) *with bounded disturbances* $|w| \leq w_m$*, to examine the closed-loop stability under the encrypted iterative distributed LMPCs of Eq.* (5.2.12) *and Eq.* (5.2.13)*. The initial state* x_0 *is assumed to be within* $\Omega_{\hat{\rho}}$*. Utilizing the results derived in Theorem 5.2.1, and preserving our earlier assumption that* $\rho > \hat{\rho} > \rho_{\min} > \rho_s$*, if the ensuing*

conditions are met,

$$\begin{aligned} &\dot{V} \leq L'_x M_F \Delta + L'_{e_1}|e_1| + L'_{e_2}|e_2| + L'_w|w| - \frac{c_3}{c_2}\rho_s \leq -\epsilon_w \\ &\rho_{\min} = \max\{V(x(t+\Delta))|V(x(t)) \leq \rho_s\} \end{aligned} \tag{5.2.23}$$

then the closed-loop state $x(t)$ remains inside $\Omega_{\hat{\rho}}$ and is ultimately bounded within $\Omega_{\rho_{\min}}$ for $t \geq T$, by implementing the encrypted iterative distributed LMPCs of Eq. (5.2.12) *and Eq.* (5.2.13).

Proof. First, we establish the feasibility of the optimization problem associated with each LMPC in the encrypted distributed LMPC system, for all the states bounded within $\Omega_{\hat{\rho}}$. Subsequently, with the optimized control inputs from the encrypted distributed LMPC, we will demonstrate that the closed-loop state of Eq. (5.2.1) is bounded and converges to the stability region $\Omega_{\hat{\rho}}$, thereby extending the findings presented from Theorem 5.2.1. If $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_{\min}}$, the input trajectories, $u_j(t)$, where $j = \{1, 2\}$ for $t \in [t_k, t_{k+1})$ are feasible solutions of the optimization problem of each LMPC, as these trajectories satisfy the constraints of Eq. (5.2.12c) and Eq. (5.2.13c), as well as the Lyapunov constraints of Eq. (5.2.12e) and Eq. (5.2.13e). Additionally, if $x(t_k) \in \Omega_{\rho_{\min}}$, the control inputs $u_j(t)$, $j = \{1, 2\}$ meet the constraints imposed in Eq. (5.2.12c) and Eq. (5.2.13c), as well as the Lyapunov constraints of Eq. (5.2.12f) and Eq. (5.2.13f); hence, the predicted states by the LMPC model are bounded within $\Omega_{\rho_{\min}}$. Thus, for all $x_0 \in \Omega_{\hat{\rho}}$, the LMPC optimization problems of Eq. (5.2.12) and Eq. (5.2.13) can be solved recursively for all iterations with feasible solutions as $x(t) \in \Omega_{\hat{\rho}}$ for all times.

Next, we establish that for any $x_0 \in \Omega_{\hat{\rho}}$, the state of the closed-loop system remains bounded within $\Omega_{\hat{\rho}}$ for all times, and given a sufficiently large time $T > 0$, it converges to a small neighborhood $\Omega_{\rho_s} \subseteq \Omega_{\rho_{\min}}$ and remains there. Under the encrypted iterative distributed LMPC system, the

time derivative of the control Lyapunov function is:

$$\dot{V} = \frac{\partial V(x(t))}{\partial x} f(x(t), \hat{u}_1(t_k), \hat{u}_2(t_k), w) \tag{5.2.24}$$

From the Lyapunov constraint of Eq. (5.2.12e) and Eq. (5.2.13e), the following inequality holds:

$$\begin{aligned} \dot{V} &= \frac{\partial V(x(t))}{\partial x} f(x(t), \hat{u}_1(t_k), \hat{u}_2(t_k), w) \\ &\leq \frac{\partial V(x(t))}{\partial x} f(x(t), \hat{\Phi}_1(\hat{x}(t_k)), \hat{\Phi}_2(\hat{x}(t_k)), w) \end{aligned} \tag{5.2.25}$$

However, extending the results of Theorem 5.2.1, the time-derivative of the control Lyapunov function under the encrypted iterative distributed LMPC can be bounded as follows:

$$\begin{aligned} &\frac{\partial V(x(t))}{\partial x} f(x(t), \hat{u}_1(t_k), \hat{u}_2(t_k), w) \leq L'_x M_F \Delta \\ &\quad + L'_{e_1}|e_1| + L'_{e_2}|e_2| + L'_w|w| - \frac{c_3}{c_2}\rho_s \leq -\epsilon_w \end{aligned} \tag{5.2.26}$$

Hence, for the selected time T, there exist positive real numbers d, Δ, and ϵ_w, such that the subsequent inequality holds $\forall\, t \in [0, T]$,

$$\dot{V} \leq L'_x M_F \Delta + L'_{e_1}|e_1| + L'_{e_2}|e_2| + L'_w|w| - \frac{c_3}{c_2}\rho_s \leq -\epsilon_w$$

which implies that $\dot{V} \leq -\epsilon_w$ for any $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_s}$ for all $t_k \in [0, T]$. This confirms that when the conditions of Eq. (5.2.23) are satisfied, the closed-loop system state remains consistently bounded within $\Omega_{\hat{\rho}}$. Furthermore, it converges to $\Omega_{\rho_s} \subseteq \Omega_{\rho_{\min}}$ within time T and stays there. With this, the proof for the stability of the system under the encrypted distributed LMPC is concluded.

□

5.2.4 Application to a nonlinear chemical process network operating at an unstable steady state

In this section, we demonstrate the application of the proposed encrypted iterative distributed LMPC system to a nonlinear chemical process that is to be operated at an unstable steady state.

5.2.4.1 Process description and model development

The process considered involves the production of ethylbenzene (EB) through the reaction of ethylene (E) and benzene (B) in two separate non-isothermal continuous stirred tank reactors (CSTRs), connected in series, as illustrated in Figure 5.2.2. The principal reaction, referred to as "primary", is a second-order, irreversible, and exothermic reaction, accompanied by two additional side reactions. The chemical reactions can be described as follows:

$$\mathrm{C_2H_4} + \mathrm{C_6H_6} \rightarrow \mathrm{C_8H_{10}} \qquad \text{(primary)} \tag{5.2.27a}$$

$$\mathrm{C_2H_4} + \mathrm{C_8H_{10}} \rightarrow \mathrm{C_{10}H_{14}} \tag{5.2.27b}$$

$$\mathrm{C_6H_6} + \mathrm{C_{10}H_{14}} \rightarrow 2\mathrm{C_8H_{10}} \tag{5.2.27c}$$

Comprehensive information on the first-principles-based dynamic model, including equations, model parameter values, and steady-state values are provided in [39]. The state variables are the concentration of ethylene, benzene, ethylbenzene, di-ethylbenzene, and the reactor temperature for each CSTR in deviation terms, that is: $x^\top = [C_{E_1} - C_{E_{1s}},\ C_{B_1} - C_{B_{1s}},\ C_{EB_1} - C_{EB_{1s}},\ C_{DEB_1} - C_{DEB_{1s}},\ T_1 - T_{1s},\ C_{E_2} - C_{E_{2s}},\ C_{B_2} - C_{B_{2s}},\ C_{EB_2} - C_{EB_{2s}},\ C_{DEB_2} - C_{DEB_{2s}},\ T_2 - T_{2s}]$.

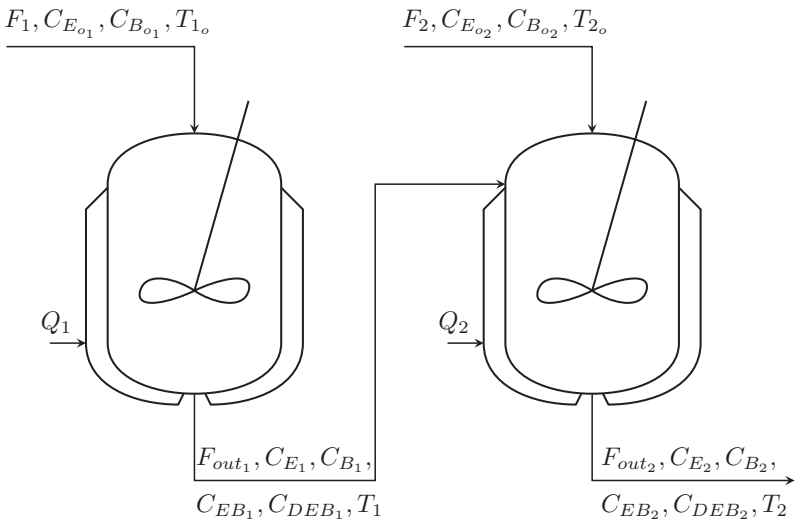

Figure 5.2.2: Process schematic of the two CSTR network.

The subscript "s" denotes the steady-state value. We create two distributed LMPCs to control the overall process. LMPC 1 optimizes the control inputs $u_1 = [C_{E_{o1}} - C_{E_{o1s}}, C_{B_{o1}} - C_{B_{o1s}}, Q_1 - Q_{1s}]^\top$. These inputs are bounded by the closed sets $[-3, 3]$ $\mathrm{kmol/m^3}$, $[-3, 3]$ $\mathrm{kmol/m^3}$, and $[-10^4, 2\times10^3]$ kW, respectively. LMPC 2 optimizes the control inputs $u_2 = [C_{E_{o2}} - C_{E_{o2s}}, C_{B_{o2}} - C_{B_{o2s}}, Q_2 - Q_{2s}]^\top$. These manipulated inputs are bounded by the closed sets, $[-2.5, 2.5]$ $\mathrm{kmol/m^3}$, $[-2.5, 2.5]$ $\mathrm{kmol/m^3}$, and $[-1.5 \times 10^4, 5 \times 10^3]$ kW, respectively. The primary goal is to manage both CSTRs at their unstable equilibrium point by utilizing the encrypted iterative distributed LMPC system. This involves the use of quantized states and control inputs for the purposes of computation and actuation.

5.2.4.2 Encrypting the distributed control system

Prior to integrating encryption and decryption into a process, the parameters d, l_1, and l_2 are chosen, considering the extreme feasible states and inputs. This involves deriving the integer bit count

$l_1 - d$. In the $\mathbb{Q}_{l_1,d}$ set, the upper limit is $2^{l_1-d-1} - 2^{-d}$, and the lower limit is -2^{l_1-d-1}. Within the set, rational numbers are separated by a resolution of 2^{-d}. The quantization parameter d, representing the fractional bit count, is determined by the desired precision level and the range of state and input values. l_2 is chosen to exceed l_1. For the case discussed in this section, $l_1 - d = 16$, and subsequently, l_1 and d are fixed. Next, $d = 8$ is chosen for simulations. Accordingly, l_1 is 24, and l_2 is 30. Encryption (Paillier cryptosystem) is implemented using Python's "phe" module, PythonPaillier [21]. To solve the multi-constrained, non-convex optimization task of the LMPCs, the IPOPT software [83] in Python is utilized.

The termination criterion for the distributed LMPCs was set to 2 iterations. Thus, control inputs are exchanged only once with the other LMPC, at the end of the first iteration. For the computation of the control cost of the distributed LMPCs, the integration step is set to $h_c = 10^{-2} \times \Delta$. We assume a control Lyapunov function of the form $V = x^\top P x$, where P is a positive definite matrix chosen as $\text{diag};[200 \;\; 200 \;\; 400 \;\; 1000 \;\; 2.5 \;\; 250 \;\; 250 \;\; 200 \;\; 1000 \;\; 0.5]$, through extensive simulations. Autocorrelated noise, represented as $w_k = 0.75 \times w_{k-1} + \xi_k$, was introduced to the inlet flow rates, F_1 and F_2, but the liquid level remains constant in both CSTRs at all times. Here, $k = 1, 2, \ldots$ denotes discrete time steps of $10^{-2} \times \Delta$, ξ_k is a randomly generated normally distributed variable with zero mean, and a standard deviation of 5% of the inlet flow rates. The prediction horizon of both LMPC is set to two sampling periods. The stability region is set as $\rho = 1800$, while $\rho_{\min} = 2$ represents the smaller region within which the closed-loop system state is desired to be bounded. The distributed LMPC cost function is defined as $L(x, u) = x^\top Q x + u^\top R u$, where $Q = \text{diag};[1000 \;\; 1000 \;\; 1500 \;\; 5 \;\; 8 \;\; 1000 \;\; 1000 \;\; 3000 \;\; 5 \;\; 110]$ and $R = \text{diag};[2.1 \;\; 1.95 \;\; 1.5 \times 10^{-5} \;\; 10 \;\; 10 \;\; 0.5 \times 10^{-4}]$. As the undesired byproduct, di-ethylbenzene,

is present in minimal quantities in both CSTRs, its trajectories are not illustrated. Non-Gaussian measurement noise obtained from the noise distribution in [57] extracted from industrial data, is added to all the measured states. As this noise is normalized, it was scaled by 2% of the operating steady-state value for each state.

It must be ensured that the sampling time (Δ) exceeds the combined time needed for encryption-decryption of the states and control inputs, along with the time required by the LMPCs to compute the control inputs at each sampling instance for the given quantization parameter. In mathematical terms,

$$\begin{aligned}\Delta > &\max(\text{Encryption-decryption time}) \\ &+ \max(\text{Control input computation time})\end{aligned} \tag{5.2.28}$$

The control inputs are applied in a sample-and-hold manner throughout the sampling period. As long as the time required for computing control inputs and encryption–decryption is shorter than the sampling period, no lag in the control variables would occur. As explained in Remark 5.2.3 and Remark 5.2.4, the time needed to encrypt–decrypt states and inputs depends on the bit lengths of the keys, number of microcontrollers, and RF modules used, and hence can be decided accordingly. For simulation purposes, 1024-bit length keys were used for encrypted communication between controllers, and 2048-bit length keys were utilized for all other encrypted communications. Considering the above criteria, assuming all encryption-decryption operations to be performed in series, although it can be done in parallel, the sampling time Δ was selected as 30 seconds in this example. Based on the constraint of Eq. (5.2.28), the encrypted distributed LMPC can only be implemented in systems that allow us to use sufficiently large sampling times that also stabilize the system as per the constraint of Eq. (5.2.26). Eq. (5.2.12e) and Eq. (5.2.13e) are Lyapunov

constraints that ensure that the time-derivative of the control Lyapunov function is more negative under the encrypted distributed LMPC than the stabilizing controller for the control input applied over the next sampling period. The future control input computed by the LMPC beyond the next sampling period may not yield a more negative time-derivative of the control Lyapunov function. Hence, we have utilized the stabilizing controller for the other subsystems in the first iteration. Moreover, as the system is operated at an unstable equilibrium, stability is critical. Alternatively, the neighboring LMPCs can utilize the future control inputs when the system is operated at a stable equilibrium.

5.2.4.3 Simulation results of the encrypted distributed LMPC system

Figure 5.2.3, Figure 5.2.4, and Figure 5.2.5 depict the results of the encrypted iterative distributed LMPC against the encrypted centralized LMPC. The normalized sum of the cost function for the encrypted distributed and centralized LMPCs was 0.9795 and 1, respectively. Also, the average computational time needed to compute the optimal control inputs by the distributed LMPC system and the centralized LMPC was 7.33 s and 13.14 s, respectively. Thus, not only did the distributed LMPC provide better closed-loop performance, but it also reduced the average computational time significantly compared to the centralized LMPC. This is evident with the fewer oscillations observed in the control input trajectories of the encrypted distributed LMPC in Figure 5.2.5. No significant difference was observed in the closed-loop state trajectories in both cases, as visible in Figure 5.2.3 and Figure 5.2.4. Nonetheless, in both cases, the system successfully converges within $\Omega_{\rho_{\min}}$ in approximately 1.5 hours of process time. We note that the time of convergence to the steady state for the desired product ethylbenzene is longer in the second CSTR as it starts with

a low initial concentration; this time may be reduced by modifying the second reactor design to adjust the residence time to speed up the second reactor dynamics.

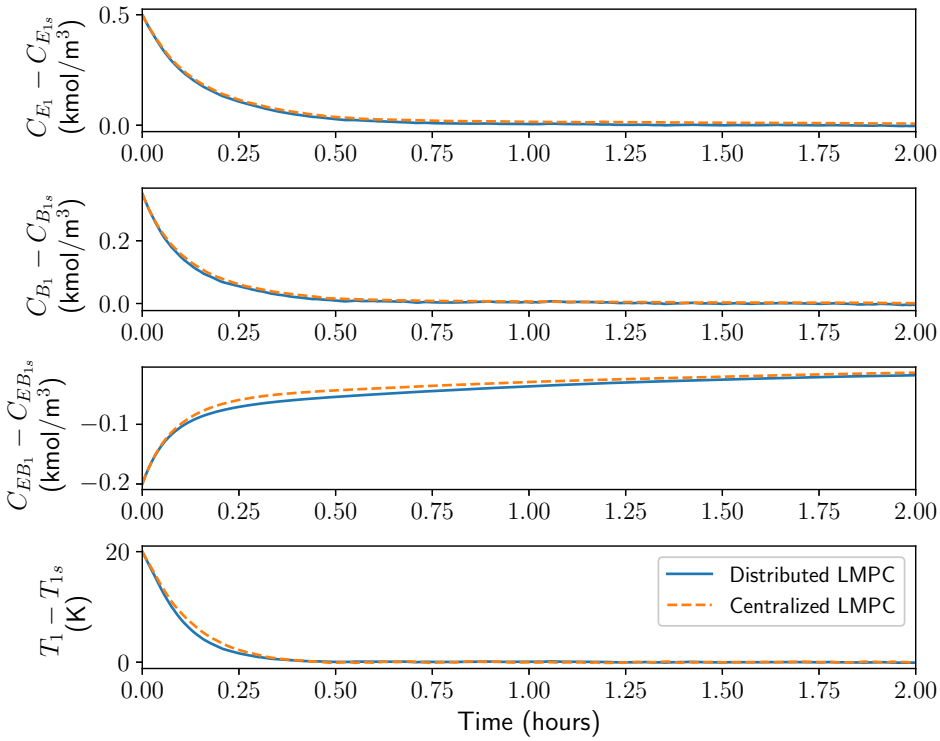

Figure 5.2.3: State trajectories of CSTR 1 under the encrypted iterative distributed LMPC (blue solid line) and encrypted centralized LMPC (orange dashed line).

Remark 5.2.8. *For the encrypted distributed LMPC investigated in this research, encryption–decryption of data as depicted in Figure* 5.2.1 *leads to errors due to quantization. [39] emphasized the potential for these errors to surpass plant/model mismatch errors in cases where distinct models are utilized in the controlled process and the LMPC. To mitigate the error caused by quantization, a higher quantization parameter* d *was recommended. Adopting* $d = 8$ *resulted in nearly indistinguishable closed-loop results with encryption when compared to those without encryption. Therefore, a quantization parameter of* $d = 8$ *was uniformly applied in all simulations conducted in this study.*

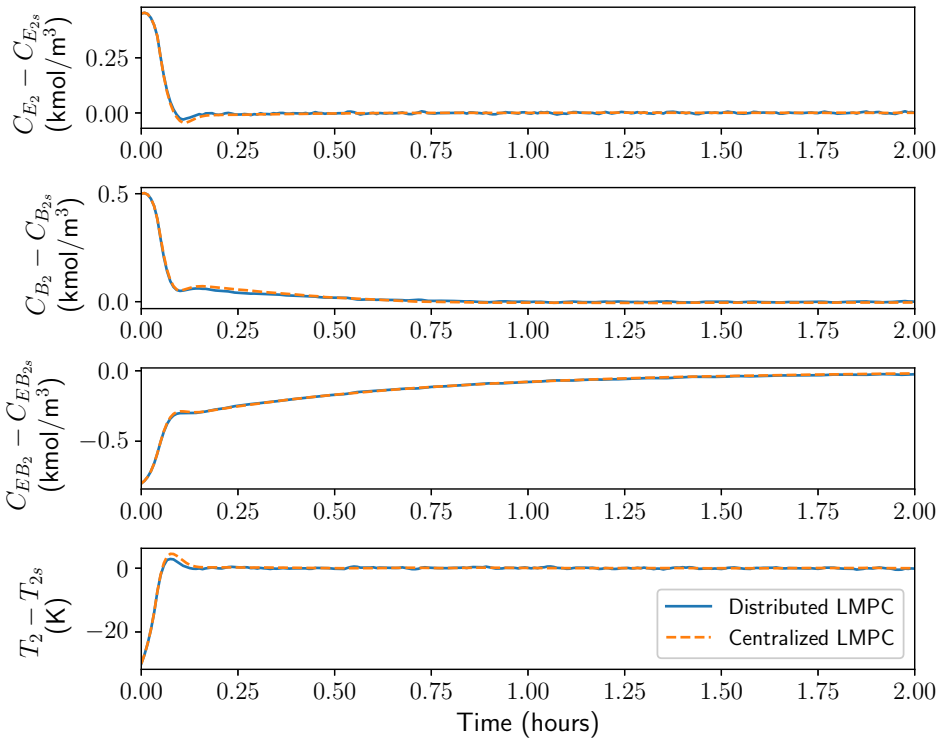

Figure 5.2.4: State trajectories of CSTR 2 under the encrypted iterative distributed LMPC (blue solid line) and encrypted centralized LMPC (orange dashed line).

5.2.5 Conclusion

In this chapter, we formulated an encrypted iterative distributed LMPC system employing encrypted signals for data transmission between sensors, controllers, and actuators. Following a comprehensive stability analysis, we determined bounds for errors from quantization, process disturbances, and the sample-and-hold implementation of the controller. With these bounds, the system could be stabilized within the desired stability region. Selection of encryption-decryption key lengths, quantization parameters, sampling time criterion, and potential methods to decrease the encryption–decryption time were discussed to facilitate practical implementation. Closed-loop simulations were performed, comparing the proposed control scheme against the encrypted centralized LMPC. Non-Gaussian sensor noise obtained from an industrial data set and process distur-

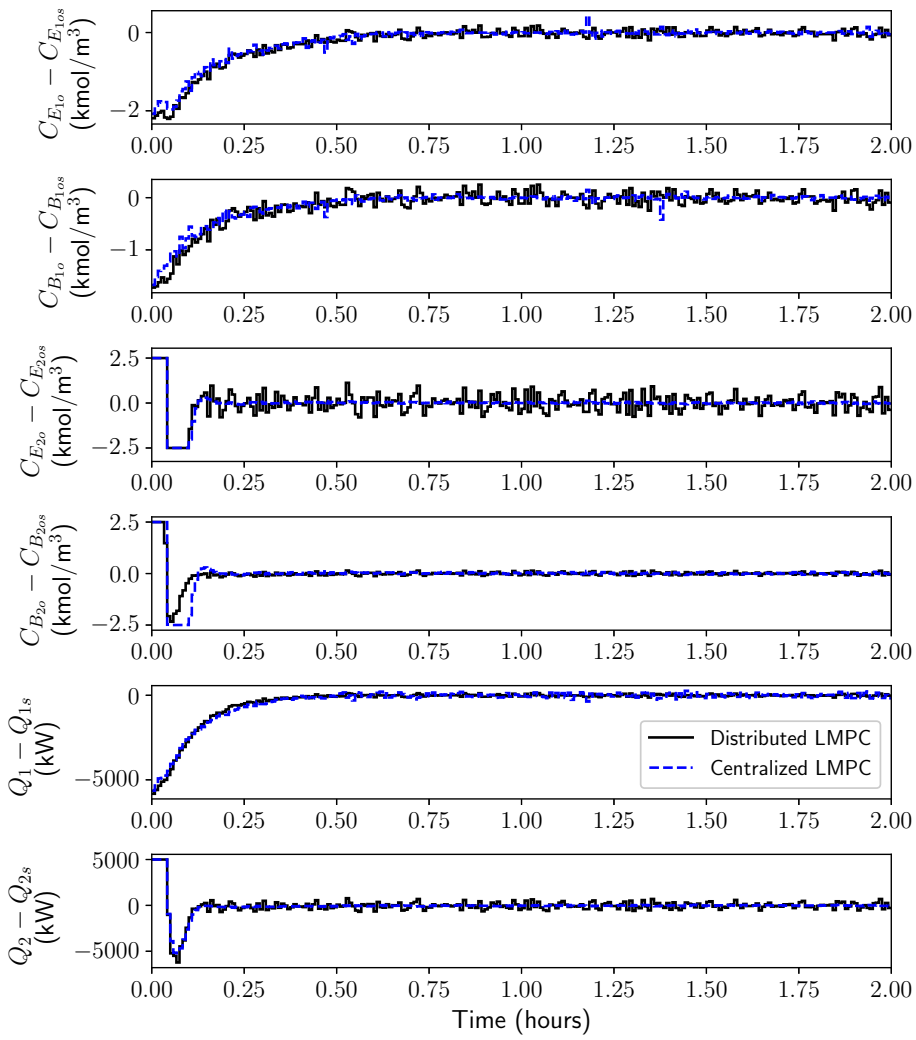

Figure 5.2.5: Control input trajectories under the encrypted iterative distributed LMPC (black solid line) and encrypted centralized LMPC (blue dashed line).

bances were used to demonstrate the industrial relevance and suitability of the proposed approach. The results favor the use of the encrypted distributed LMPC system, which not only improves closed-loop performance but also significantly reduces the computational time needed to calculate the control input, positioning the encrypted iterative distributed LMPC as an effective solution for improving closed-loop performance, decreasing computational time, and enhancing cybersecurity in large-scale nonlinear systems.

Chapter 6

Integrating dynamic economic optimization and encrypted control for cyber-resilient operation of nonlinear processes

6.1 Introduction

Networked control systems have emerged as a transformative paradigm in industrial operations, offering numerous advantages [90]. By harnessing networked communication protocols, these systems significantly reduce the need for extensive wiring and hardware, leading to cost savings and streamlined operations. Additionally, they modernize plant infrastructure by enabling real-time monitoring and control, thereby enhancing operational efficiency and responsiveness. With fewer physical components, maintenance issues are minimized, contributing to improved system reliability and reduced downtime. Further, the ease of implementation and scalability make networked control systems accessible to a wide range of applications, from small-scale operations

to large industrial complexes. Given these benefits, networked control systems have become the standard for control systems, offering unparalleled flexibility, efficiency, and reliability in managing industrial processes. As technology continues to evolve, embracing networked control systems remains imperative for organizations aiming to maintain competitiveness and adaptability in an ever-changing industrial landscape.

While networked communication facilitates seamless and rapid data transfer, it also introduces vulnerabilities to cyberthreats. Breaches or compromises in these systems can have severe consequences such as disruptions of essential services or physical harm, which are threats to public safety. Recent advances in cyberattack techniques support the imperative of establishing robust cybersecurity protocols [31]. Real-world incidents reaffirm the critical need of cybersecurity in networked cyber-physical systems. For example, the 2015 BlackEnergy malware attack on SCADA controls overseeing Ukraine's power grid resulted in widespread power outages [45]. Similarly, Colonial Pipeline suffered a ransomware attack by DarkSide hackers in 2021, when its networked communication was encrypted and a ransom was demanded for the decryption keys. Subsequently, Colonial Pipeline had to shut down its fuel distribution operations, resulting in significant financial losses [82]. As cyber threats continue to evolve, cybersecurity concerns loom over process control systems. Modern control systems must be designed with robust security measures to mitigate the impact of cyberattacks. Some measures include implementing secure communication protocols, regularly updating software and firmware, and employing cyberattack detection systems with reconfiguration protocols in the event of an attack.

In traditional process control frameworks, model predictive control (MPC) is combined with a real-time optimizer (RTO), the latter of which is tasked with determining economically opti-

mal steady-states to be tracked by the MPC through a comprehensive plant model. However, as energy consumption and operational efficiency concerns escalate in industries like chemical and petrochemicals, economic model predictive control (EMPC) has emerged. EMPC enables dynamic optimization of economic cost functions while maintaining stability constraints. Extensive research in chemical process control literature indicates that several industrial processes can attain greater profits through time-varying operation compared to constant steady-state operation [6, 29]. Also, today's dynamic economic landscape is characterized by rapid globalization, technological advancements, and unforeseen disruptions. Fluctuations in energy costs, commodity prices, currency values, interest rates, logistics costs, and market trends can significantly impact businesses and industries worldwide. By incorporating fluctuating real-world economics, EMPC systems can yield superior results, emphasizing the importance of dynamic optimization techniques for maximizing economic benefits and maintaining competitiveness in volatile environments.

Previous studies have explored topics like implementing secure communication in networked control systems through encryption [19, 20], developing cyberattack detectors [2, 26], creating cyberattack-resilient controllers [68], and developing economic MPCs with time-varying objective functions [28, 29]. However, these efforts have not yet resulted in control systems resilient to cyber threats that seamlessly integrate secure communication, cyberattack detection, nonlinear dynamic economic optimization, and real-time fluctuations in economics. Establishing such capabilities is critical for contemporary control systems to navigate economic challenges and cyber vulnerabilities in dynamic environments. This gap motivates our proposal for a new control framework aimed at effectively addressing this challenge.

Specifically, we introduce an encrypted two-layer control framework comprising a nonlin-

ear Lyapunov-based economic model predictive control (LEMPC) scheme in the upper layer and an encrypted linear feedback control system in the lower layer. As we cannot perform nonlinear computations in an encrypted space, we decrypt state information in the upper layer to determine the economically optimal dynamic set point trajectory via nonlinear optimization. Conversely, the lower layer securely tracks these set points in an encrypted space without decryption, utilizing the additive homomorphic property of the Paillier cryptosystem for secure, private communication. To address the cyber vulnerability of the upper layer, we integrate a logic-based cyberattack detector. In the event of an attack, the encrypted lower layer autonomously continues operation, disregarding compromised signals from the upper layer, thus ensuring cyber-resilient operation. In [29], a two-level EMPC system was implemented, consisting of an EMPC in the upper level computing the operating trajectory for the lower-level LMPC to track through closed-loop feedback. In our framework, the objective is to facilitate encrypted operating trajectory tracking without decryption using encrypted feedback at the lower-layer, by employing proportional-integral (PI) controllers which allow linear mathematical operations to be performed in an encrypted space without decryption. Unlike the previous approach which lacked encryption, this method ensures secure communication. While utilizing LMPC in the lower layer would enhance control input computation optimization, it would not fortify against cyberattacks as the computations would occur without encryption.

The subsequent sections of the paper are structured as follows: in Section 6.2, we cover preliminaries, including notation, the class of systems under consideration, system stability assumptions, the cryptosystem employed for encryption, and the effects of quantization. In Section 6.3, we discuss the encrypted two-layer control framework, formulate the LEMPC, and present the stabil-

ity analysis of the proposed control system. Section 6.4 presents and analyzes various closed-loop simulations of a nonlinear chemical process within the encrypted two-layer control framework.

6.2 Preliminaries

6.2.1 Notation

The notation $x^\top$ represents the transpose of a vector x. The sets of real numbers, integers, and natural numbers are represented by $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$, respectively. Additionally, $\mathbb{Z}_M$ refers to the additive group of integers modulo M. Set subtraction is indicated by "\", where $A \backslash B$ denotes the set of elements in A but not in B. A function denoted by $f(\cdot)$ belongs to the class $\mathcal{C}^1$ if it is continuously differentiable within its domain. Furthermore, a continuous function $\alpha : [0, a) \to [0, \infty)$ is classified as class $\mathcal{K}$ if $\alpha(0) = 0$, and it is strictly increasing. The terms $\mathrm{lcm}(i, j)$ and $\gcd(i, j)$ represent the least common multiple and greatest common divisor of integers i and j, respectively.

6.2.2 Class of systems

In this research, we focus on multi-input multi-output (MIMO) nonlinear systems, which are described by a set of ordinary differential equations (ODEs) in the following manner:

$$\begin{aligned} \dot{x} &= f(x(t), u(t), w(t)) \\ y &= x + v \end{aligned} \tag{6.1}$$

The state vector is represented by $x \in \mathbb{R}^n$, and $y \in \mathbb{R}^n$ denotes the vector of continuously sampled state measurements. The control input vector, denoted by $u \in \mathbb{R}^m$, is subject to bounds defined by the set $U \subset \mathbb{R}^m$. Specifically, U is defined as $U = \{u \in \mathbb{R}^m | u_i^{\min} \leq u_i \leq u_i^{\max}, i = 1, \ldots, m\}$, where $u_i^{\min}$ and $u_i^{\max}$ represent the lower and upper bounds, respectively, of the i^{th} control input in the vector u. Additionally, the disturbance vector is denoted by $w \in \mathbb{R}^w$, and the noise vector is denoted by $v \in \mathbb{R}^n$. Similarly, the disturbance and noise vectors are bounded by $|W(t)| \leq \theta$ and the set $\bar{V} \in \mathbb{R}^n$, respectively. The function $f(\cdot)$ is locally Lipschitz and evaluates to zero at the origin $f(0, 0, 0) = 0$, establishing it as an equilibrium of Eq. (6.1). We set the initial time as zero ($t_0 = 0$). Further, $S(\Delta)$ is defined as the set of piece-wise constant functions with a period of Δ.

We introduce a dynamic economic optimization and encrypted feedback control framework to guide the system of Eq. (6.1) in tracking the reference trajectory representing time-varying operating set points, $x_E(t) \in \Omega_\rho$, where Ω_ρ is defined in the subsequent subsection. The rate of change of the reference trajectory is bounded by:

$$|\dot{x}_E(t)| \leq \gamma_E \tag{6.2}$$

To capture the deviation between the actual state trajectory $x(t)$ and the time-varying reference trajectory $x_E(t)$, we introduce,

$$e(t) = x(t) - x_E(t) \tag{6.3}$$

and we can characterize its dynamics by

$$\begin{aligned}\dot{e} &= \dot{x}(t) - \dot{x}_E(t) \\ &= f(x(t), u(t), w(t)) - \dot{x}_E(t) \\ &= f(e(t) + x_E(t), u(t), w(t)) - \dot{x}_E(t) \\ &= g(e(t), x_E(t), \dot{x}_E(t), u(t), w(t)).\end{aligned} \tag{6.4}$$

We assume that Eq. (6.4) is continuously differentiable and possesses a unique equilibrium point for each fixed $x_E \in \Omega_\rho$. In other words, for every x_E there exists a corresponding u_E, resulting in $e = 0$ being an equilibrium of Eq. (6.4). This condition can be expressed mathematically as

$$g(0, x_E, 0, u_E, 0) = 0 \tag{6.5}$$

Remark 6.1. *Assuming that the system described by Eq.* (6.1) *has an equilibrium for each fixed* $x_E \in \Omega_\rho$ *is crucial for enabling the tracking of the reference trajectory. With an economic model predictive controller (EMPC) in place, the economically optimal dynamic state trajectory can be determined for any initial condition* $x_E(t_0) \in \Omega_\rho$*, where* $t_0 = 0$*. Consequently, the generated reference state trajectory contains set points that can be effectively tracked for any* $x_E \in \Omega_\rho$*.*

6.2.3 Stability assumptions

We assume the existence of an explicit stabilizing feedback control law, $u(t) = h(e(t), x_E(t)) \in U$, that renders the origin of the system of Eq. (6.1) with $w \equiv 0$ and $v \equiv 0$ asymptotically stable, for each $x_E \in \Omega_\rho$. This assumption guarantees that the time-varying state trajectory $x_E(t)$ can be

tracked and signifies the existence of a $\mathcal{C}^1$ control Lyapunov function $V(e, x_E)$ that satisfies the following inequalities:

$$\alpha_1(|e|) \leq V(e, x_E) \leq \alpha_2(|e|), \tag{6.6a}$$

$$\frac{\partial V(e, x_E)}{\partial e} g(e, x_E, 0, h(e, x_E), 0) \leq -\alpha_3(|e|), \tag{6.6b}$$

$$\left| \frac{\partial V}{\partial e} \right| \leq \alpha_4(|e|), \tag{6.6c}$$

$$\left| \frac{\partial V}{\partial x_E} \right| \leq \alpha_5(|e|) \tag{6.6d}$$

$\forall e, x_E \in \mathbb{R}^n$ in an open region D surrounding the origin. The functions $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, and α_5 belong to the class $\mathcal{K}$. For the system of Eq. (6.1), the region of closed-loop stability can be defined as a level set denoted by Ω_ρ of the control Lyapunov function V. This set is described as $\Omega_\rho := \{x \in D | V(e, x_E) \leq \rho\}$, where $\rho > 0$. Therefore, starting from any initial condition inside Ω_ρ, the control input $h(e, x_E)$ ensures that the closed-loop state trajectory remains within Ω_ρ.

Based on the continuity of f, we attribute the local Lipschitz property to the vector field f. Further, considering that the manipulated input vector u is bounded within nonempty convex sets, a positive constant exists such that

$$|f(x, u, w)| \leq M_F \tag{6.7}$$

$\forall x \in \Omega_\rho$, $u \in U$, and $w \in W$. Extending this to the system of Eq. (6.4), considering that the rate of change of $x_E(t)$ is bounded by γ_E,

$$|g(e, x_E, \dot{x}_E, u, w)| \leq M \tag{6.8}$$

$\forall(x - x_E) \in \Omega_{\rho^*}$, $x_E \in \Omega_\rho$, $u \in U$, and $w \in W$. Further, due to the continuous differentiability of the control Lyapunov function $V(e, x_E)$ and the Lipschitz property of f, there exist positive constants $L_w, L'_w, L_e, L'_e, L'_E, L''_E, L'_u$ such that

$$|g(e, x_E, \dot{x}_E, u, w)| - |g(e', x'_E, \dot{x}_E, u, 0)| \leq L_e|e - e'| + L_E|x_E - x'_E| + L_w|w|, \tag{6.9}$$

$$\begin{aligned}\left|\frac{\partial V(e, x_E)}{\partial e} g(e, x_E, \dot{x}_E, u, w) - \frac{\partial V(e', x'_E)}{\partial e} g(e', x'_E, \dot{x}'_E, u', 0)\right| \leq\ & L'_e|e - e'| + L'_E|x_E - x'_E| \\ & + L''_E|\dot{x}_E - \dot{x}'_E| + L'_w|w| \\ & + L'_u|u - u'|\end{aligned} \tag{6.10}$$

$\forall x_E, x'_E \in \Omega_\rho$, $e, e' \in \Omega_{\rho^*}$, $|\dot{x}_E| \leq \gamma_E$, $|\dot{x}'_E| \leq \gamma_E$, $u \in U$, and $w \in W$.

Remark 6.2. *In various nonlinear systems commonly encountered in chemical process control systems, Lyapunov functions have often been formulated using state variables $V(x) = \bar{f}(x(t))$. In our study, leveraging the previous definitions of e and x_E, we can also represent the state vector as $x(t) = x_E(t) - e(t)$. Consequently, we broaden the Lyapunov function to take the form $V(e, x_E)$, as we proceed to examine the stability of the system within the proposed control framework in the following section.*

6.2.4 Paillier cryptosystem

In this study, we utilize the Paillier cryptosystem [67] to encrypt various signals, including state measurements (y), reference trajectory set points (x_E), and manipulated inputs (u), which are transmitted to and from the controllers. A key aspect of our approach is utilizing the semi-homomorphic property of additive homomorphism inherent in the Paillier cryptosystem. This property enables

us to perform linear additive operations in an encrypted space, particularly within the lower encrypted feedback control layer. The encryption process begins with the generation of both public and private keys. As the Paillier cryptosystem is an asymmetric encryption scheme, it utilizes two different keys for encryption and decryption: a public key for encrypting plaintext and a private key for decrypting ciphertext. The procedure for generating these keys is:

1. Select two large prime integers (p and q) based on the desired key bit length, such that, $\gcd(pq, (p-1)(q-1)) = 1$.

2. Calculate, $M = pq$.

3. Search for an arbitrary integer $\bar{g}$ such that $\bar{g} \in \mathbb{Z}_{M^2}$, that is, the multiplicative group of integers modulo M^2.

4. Calculate $\lambda = \mathrm{lcm}(q-1, p-1)$.

5. Define $\bar{L}(x) = (x-1)/M$.

6. Verify the existence of the subsequent modular multiplicative inverse:
   $u = (\bar{L}(\bar{g}^{\lambda} \bmod M^2))^{-1} \bmod M$.

7. If the inverse does not exist, go back to step 3. If the inverse exists, $(M, \bar{g})$ is the public key and (λ, u) is the private key.

After obtaining the keys, authorized recipients receive the public and private keys for encryption and decryption, respectively. The message m is encrypted as follows:

$$E_M(m, r) = c = \bar{g}^m r^M \bmod M^2 \tag{6.11}$$

where r is a random integer from the set $\mathbb{Z}_M$, and c is the resulting ciphertext obtained by encrypting m. Decryption is performed as follows to obtain m:

$$D_M(c) = m = \bar{L}(c^{\lambda} \bmod M^2)u \bmod M. \tag{6.12}$$

6.2.5 Quantization

Prior to encrypting data using the Paillier cryptosystem, it must be processed to natural numbers in $\mathbb{Z}_M$. However, signal values are typically in floating-point format before encryption. As a result, a process known as quantization is employed to convert the floating-point numbers into $\mathbb{Z}_M$ [19]. This involves creating a set, denoted as $\mathbb{Q}_{l_1,d}$, which is characterized by two parameters: l_1, representing the total bit count (combining integer and fractional bits), and d, indicating the number of fractional bits. The set, $\mathbb{Q}_{l_1,d}$, comprises rational numbers ranging from -2^{l_1-d-1} to $2^{l_1-d-1}-2^{-d}$, with intervals of 2^{-d}. A rational number q within $\mathbb{Q}_{l_1,d}$ can be expressed as $q \in \mathbb{Q}_{l_1,d}$, where $\exists \beta \in \{0,1\}^{l_1}$, and $q = -2^{l_1-d-1}\beta_{l_1} + \sum_{i=1}^{l_1-1} 2^{i-d-1}\beta_i$. The function, $g_{l_1,d}$ maps a real number data point a to $q \in \mathbb{Q}_{l_1,d}$ as follows:

$$\begin{aligned} g_{l_1,d} &: \mathbb{R} \rightarrow \mathbb{Q}_{l_1,d} \\ g_{l_1,d}(a) &:= \arg \min_{q \in \mathbb{Q}_{l_1,d}} |a - q| \end{aligned} \tag{6.13}$$

Subsequently, we convert the quantized data to a set of positive integers using a one-to-one (bijective) mapping referred to as $f_{l_2,d}$, as described in [19]. This mapping is structured to ensure that the

quantized data is translated into a subset of the message space, $\mathbb{Z}_M$, and is performed as follows:

$$\begin{aligned} f_{l_2,d} &: \mathbb{Q}_{l_1,d} \to \mathbb{Z}_{2^{l_2}} \\ f_{l_2,d}(q) &:= 2^d q \bmod 2^{l_2} \end{aligned} \tag{6.14}$$

In the encryption process, plaintext messages from the set $Z_{2^{l_2}}$ are transformed to ciphertexts, which can subsequently be decrypted back into the original set $Z_{2^{l_2}}$. Next, to retrieve the original data from the set $\mathbb{Q}_{l_1,d}$, an inverse mapping, labeled as $f_{l_2,d}^{-1}$, is performed as follows:

$$f_{l_2,d}^{-1} : \mathbb{Z}_{2^{l_2}} \to \mathbb{Q}_{l_1,d} \tag{6.15}$$

$$f_{l_2,d}^{-1}(m) := \frac{1}{2^d} \begin{cases} m - 2^{l_2} & \text{if } m \geq 2^{l_2-1} \\ m & \text{otherwise} \end{cases} \tag{6.16}$$

Remark 6.3. *Quantization-related errors tend to accumulate in multiplicatively homomorphic encryption schemes like ElGamal, due to the compounding nature of multiplication and associated scaling operations. In contrast, additive homomorphism, like in the Paillier scheme employed in our work, is generally less prone to quantization error accumulation, as addition does not involve scaling or compounding of errors through multiplication. To mitigate this effect, one can select a higher quantization parameter.*

6.3 Development of the encrypted two-layer control framework

In this section, we describe the design of the proposed encrypted two-layer control framework, formulate the LEMPC and encrypted feedback controller, and perform a stability analysis of the

encrypted control system.

6.3.1 Design and implementation

In the encrypted control framework illustrated in Figure 6.1, at time t_k, sensor signals $x(t_k)$ undergo encryption to form ciphertext c_1 using a public key. These encrypted signals are then transmitted to a cloud hardware security module (HSM), a dedicated hardware device utilized for managing cryptographic keys and securely performing cryptographic operations within a cloud computing environment. After decryption using the private key, the quantized sensor signals $\hat{x}(t_k)$ are sent to the cloud server responsible for nonlinear EMPC computations, aimed at determining economically optimal dynamic set points $x_E(t)$ for $t = [t_k, t_k + t')$, where t' represents the EMPC operating period. Following this, the set points $x_E(t)$ are encrypted into ciphertext c_1' using the public key within another cloud HSM. Subsequently, these encrypted set points are transmitted to a set of PI (proportional-integral) controllers in the encrypted lower feedback control layer. Operating with a sampling period Δ significantly smaller than the operating period t', this lower layer computes control inputs to track the set point trajectory using encrypted sensor signals c_2, sampled at intervals of Δ. These control input computations take place within an encrypted space without decryption, leveraging the additive homomorphic property of the Paillier cryptosystem. At the actuator, encrypted control inputs c_3 are decrypted to obtain the quantized input $\hat{u}(t_k)$, which is then applied to the process. This cycle within the lower layer continues until it receives a new encrypted state trajectory from the EMPC in the upper layer at the end of the operating period.

The closed-loop design depicted in Figure 6.1 has three potential points vulnerable to cyber-attacks for data manipulation: the updated economic information containing the weights of the

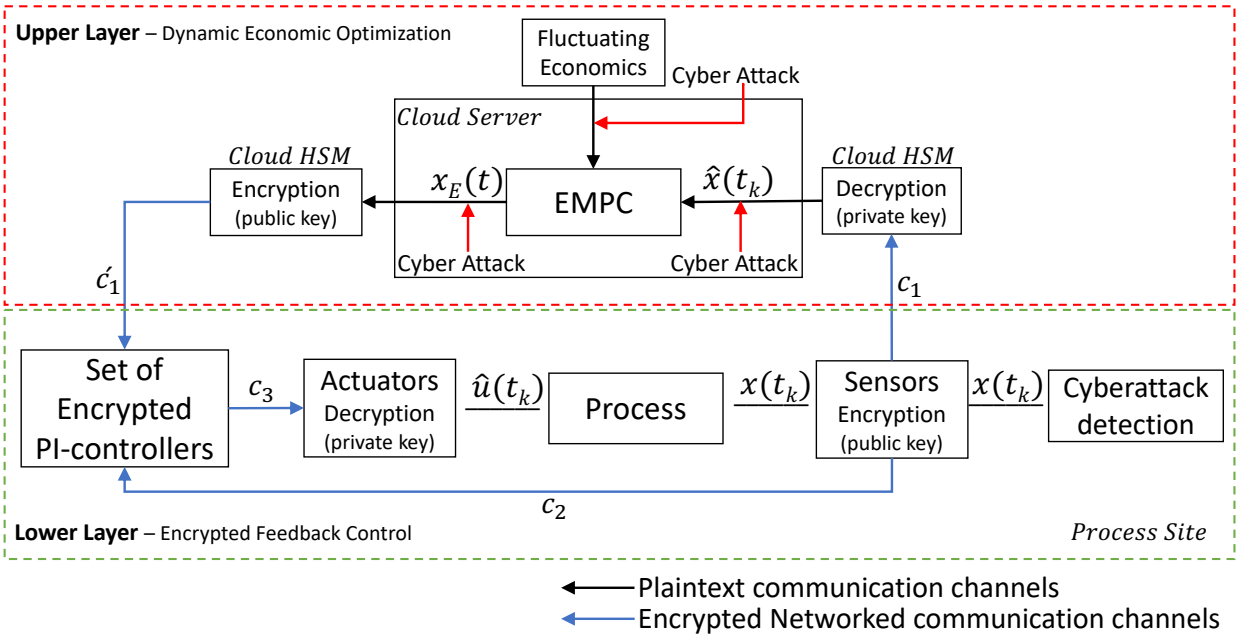

Figure 6.1: A block diagram of the proposed encrypted two-layer control framework.

EMPC objective function, the decrypted sensor signal received from the cloud HSM, and the computed set points of the EMPC before transmission to the cloud HSM. To detect potential threats initiated against the vulnerable upper layer, a logic-based cyberattack detector is integrated into the lower layer, which utilizes sensor-derived data for attack detection. Upon detection, the control system logic reconfigures, disregarding signals received from the compromised upper layer, and operates independently. Detailed information of the cyberattack detector and reconfiguration mechanism is provided in Section 6.4.

Further, this design introduces three sources of error: one stemming from state quantization in the sensor-to-upper layer EMPC link, another arising from set point quantization in the upper layer EMPC-to-lower layer feedback controller link, and a third originating from control input

quantization in the lower layer feedback controller-to-actuator link. These errors are bounded by:

$$|x(t_k) - \hat{x}(t_k)| \leq 2^{-d-1} \tag{6.17a}$$

$$|x_E(t_k) - \hat{x}_E(t_k)| \leq 2^{-d-1} \tag{6.17b}$$

$$|u(t_k) - \hat{u}(t_k)| \leq 2^{-d-1} \tag{6.17c}$$

The bounds of the quantization error, as detailed in Eq. (6.17), are derived in Remark 6.5. Further, an additional error is introduced in the applied control input. This stems from the lower layer feedback controller, $h(e, x_E)$, that uses the quantized error $\hat{e} = \hat{x} - \hat{x}_E$ to compute control inputs in an encrypted space. This error will be bounded by:

$$\begin{aligned} |e - \hat{e}| &= |(x - x_E) - (\hat{x} - \hat{x}_E)| \\ &= |(x - \hat{x}) + (\hat{x}_E - x_E)| \\ &\leq 2^{-d-1} + 2^{-d-1} \\ &\leq 2^{-d} \end{aligned} \tag{6.18}$$

Remark 6.4. *The two-layer encrypted dynamic optimization and control framework outlined in our work is adaptable and can be applied when other dynamic optimization strategies are used in the upper-layer to calculate the set points (current values of the operating trajectory) of the lower-layer control system, not just the economic MPC scheme employed in our work. The key objective of this structure is to facilitate nonlinear control and optimization within an encrypted system. In this framework, the upper layer computes set points through nonlinear dynamic optimization (which cannot be performed in an encrypted space), then encrypts these set points and transmits*

them to the lower layer. The lower layer, without decrypting the set points, utilizes encrypted measurement feedback to track these set points, integrating encryption with nonlinear dynamic optimization and control.

Remark 6.5. *Quantization error occurs when a value intended for quantization does not precisely match any value in the set $\mathbb{Q}_{l_1,d}$, which is spaced apart by 2^{-d}. Suppose the value to be quantized is denoted as a, which is positioned between b and $b + 2^{-d}$. If the absolute difference between a and b is smaller than that between a and $b + 2^{-d}$, then a is assigned to b; otherwise, it is assigned to $b + 2^{-d}$. As a result, the maximum potential difference between the actual and quantized values is half the resolution, or 2^{-d-1}. Therefore, increasing the value of d reduces the quantization error.*

Remark 6.6. *We operate the proposed closed-loop design under a few assumptions. Firstly, we assume that plaintext data is vulnerable to cyberattacks, wherein it can be manipulated or subjected to denial-of-service (DOS) attacks. However, we do not consider attacks on encrypted data due to its inherent privacy. Each encryption process generates a unique ciphertext due to the random number generated, making manipulation easily detectable. In case of an attack on encrypted data, the only recourse is to transfer control to a secure backup system isolated from any network. Secondly, we assume that the cloud server where nonlinear computations occur in plaintext is vulnerable to cyber threats. Lastly, we assume that the cloud HSMs responsible for housing cryptographic keys and performing cryptographic operations are fully secure. Cloud HSMs, offered by leading providers such as Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP), adhere to stringent security standards like FIPS 140-2/3 [12]. They are chosen precisely because they are impervious to cyberattacks, validating this assumption.*

Remark 6.7. *While the proposed closed-loop design of the encrypted two-layer control framework is vulnerable to cyberattacks, it enhances cybersecurity by integrating a cyberattack detection and subsequent reconfiguration mechanism. Furthermore, it improves the robustness of the control system by transmitting data only once during each operating period between the lower and upper control layers, reducing the potential for attacks due to less frequent data transmission. Additionally, in this design, the cloud server does not have access to either key, and no component has access to both the public and private keys; they only have access to one or the other. Also, following the prevailing standard recommended by NIST, it is recommended to use cryptographic keys with a bit-length greater than 2048 to assure robustness [9].*

Remark 6.8. *For large-scale processes with numerous states and inputs, employing a centralized MPC in the cloud server would entail significant computational expenses. Alternatively, decentralized and distributed MPCs could be integrated into the same framework to alleviate the computational burden associated with the centralized approach, as demonstrated in prior works [37, 38]. In these works, encrypted data was decrypted at each sampling instance within the nonlinear MPC to compute control inputs. However, in our approach, decryption only occurs within the LEMPC of the upper layer at the start of each operating period, rather than at every sampling period. Control inputs for tracking the reference trajectory are then computed without decryption. As a result, the frequency of encryption-decryption operations at the controllers is substantially reduced in our proposed framework. This reduction enhances security by minimizing the opportunities for manipulating decrypted data.*

Remark 6.9. *The duration of the operating period t' is established by considering the lowest fre-*

quency required for updating economic data, including energy prices, raw material costs, product demand, or product selling prices. Within the EMPC objective function, this economic information remains constant throughout the operating period. Additionally, the chosen period can be shorter than the interval between updates of economic information. In this scenario, economic data would remain constant between operating periods. However, it should still be long enough to compute state trajectories optimized over a period significantly larger than the sampling period of the lower feedback layer, where these trajectories are tracked.

6.3.2 Dynamic economic optimization

The optimization problem for the LEMPC in the upper layer of the proposed control framework is represented as:

$$\mathcal{J} = \max_{u_E \in S(\Delta_E)} \int_{t_k}^{t_{k+N_E}} L(\tilde{x}_E(t), u_E(t))\, \mathrm{d}t \tag{6.19a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}_E(t) = f(\tilde{x}_E(t), u_E(t)) \tag{6.19b}$$

$$u_E \in U,\ \forall\, t \in [t_k, t_{k+N_E}) \tag{6.19c}$$

$$|\dot{x}_E(t)| \leq \gamma_E,\ \forall\, t \in [t_k, t_{k+N_E}) \tag{6.19d}$$

$$\tilde{x}_E(t_k) = \hat{x}(t_k) \tag{6.19e}$$

$$V(\tilde{x}_E(t_k)) \leq \rho_{secure},\ \forall\, t \in [t_k, t_{k+N_E}),$$

$$\text{if } \tilde{x}_E(t_k) \in \Omega_{\rho_{secure}} \tag{6.19f}$$

$$\dot{V}(\tilde{x}_E(t_k), u_E) \leq \dot{V}(\tilde{x}_E(t_k), \Phi(\tilde{x}_E(t_k))),$$

$$\text{if } \tilde{x}_E(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{secure}} \tag{6.19g}$$

where Δ_E is the LEMPC sampling period. Eq. (6.19e) uses the quantized state, $\hat{x}(t_k)$, after decryption, to initialize the LEMPC plant model of Eq. (6.19b). k represents the sampling instance, and N_E represents the number of sampling periods within the LEMPC prediction horizon. $\tilde{x}_E(t)$ is the predicted state trajectory of the LEMPC model of Eq. (6.19b). This model is utilized to integrate the economic objective function of Eq. (6.19a) to calculate the optimized LEMPC control inputs, $u_E(t)$, where $t \in [t_k, t_k + N_E)$. The LEMPC's goal is to maximize this objective function over the prediction horizon such that it satisfies the constraints of Eqs. (6.19c) to (6.19g). Eq. (6.19c) represents the constraints imposed on the control inputs. The constraint of Eq. (6.19d) ensures that the lower layer can track the reference trajectory $x_E(t)$ by limiting its rate of change, $\dot{x}_E(t)$. From the Lyapunov constraint of Eq. (6.19f), the LEMPC ensures that, if the state $\tilde{x}(t_k) \in \Omega_{\rho_{secure}}$ at time t_k, then it lies within this region for $t \in [t_k, t_k + N_E)$, where ρ_{secure} is a level set of the control Lyapunov function $V(\tilde{x}_E)$ such that $V(\tilde{x}_E) \leq \rho_{secure}$. If $\tilde{x}_E(t_k)$ lies within the set $\Omega_\rho \setminus \Omega_{\rho_{secure}}$, the Lyapunov constraint of Eq. (6.19g), ensures that LEMPC drives the predicted state trajectory $\tilde{x}_E(t)$ to the origin at a rate faster than or at least equal to the stabilizing controller $\Phi(\tilde{x}_E(t_k))$ (the existence of $\Phi(\cdot)$ follows from the stabilizability assumption on the process made in Section 6.2.3). Following the computation of optimized control inputs u_E by LEMPC, the reference trajectory $x_E(t)$ is derived by recursively solving the model described in Eq. (6.19b), where u_E is implemented in a sample-and-hold fashion. The x_E values are logged at intervals of Δ, denoting the lower layer's sampling period, and subsequently relayed to the cloud HSM for encryption prior to transmission to the encrypted lower-layer control system for tracking.

Remark 6.10. *The proposed LEMPC operates on feedback, as it starts with actual state measure-*

ments. However, in case of an event like a denial-of-service (DOS) attack where the threat actor blocks the decrypted sensor measurements from reaching the upper layer, we can initialize the LEMPC using the final value of the predicted state trajectory from the previous operating period. This assumes that at the end of the previous operating period, the deviation between the actual state trajectory and the reference trajectory is within the bounds as derived in Section 6.3.4.

6.3.3 Encrypted feedback control

In the encrypted space, only linear mathematical operations are permissible. Consequently, we utilize the recursive rule to approximate integral terms within the set of proportional-integral controllers of the encrypted lower layer feedback control system, ensuring strictly linear mathematical operations, as illustrated below:

$$\begin{aligned} u_i(t_k) &= K_{c_i}\left(e_i(t_k) + \frac{1}{\tau_i}\int_0^{t_k} e_i(\tau)\,\mathrm{d}\tau\right) \\ &= K_{c_i}e_i(t_k) + I_{t_k} \\ &= K_{c_i}e_i(t_k) + K'_{c_i}e_i(t_k) + I_{t_{k-1}} \end{aligned} \tag{6.20}$$

where $u_i(t_k)$ is the i^{th} control input of the lower layer. The error of the i^{th} state at time t_k is described by $e_i(t_k) = x_{E_i}(t_k) - x_i(t_k)$, with $x_{E_i}(t_k)$ and $x_i(t_k)$ denoting the set point and state measurement of the i^{th} state, at time t_k, respectively. t_k and t_{k-1} denote the sampling instances k and $k-1$, respectively. K_{c_i} and K'_{c_i} represent the proportional and integral gains, while I_{t_k} denotes the integral control action at t_k. At $k = 0$, I_{t_0} is assumed to be 0. The lower layer has a sampling period of Δ, and applies the computed control inputs in a sample-and-hold manner for the time $t = [t_k, t_k + \Delta)$, and then recomputes the control input with the updated set point and

state feedback at time $t = t_{k+1}$.

6.3.4 Stability analysis

In this subsection, we examine the closed-loop stability of the proposed two-layer encrypted control framework, with the LEMPC at the upper layer and the encrypted feedback controller at the lower layer.

Theorem 6.1. *Considering the nonlinear system described in Eq.* (6.1)*, we analyze its stability under the encrypted lower layer feedback controller* $\hat{h}(\hat{e}, \hat{x}_E)$*, under the influence of bounded disturbances. The lower layer feedback controller* $h(e, x)$*, operating without encryption, satisfies the inequalities specified in Eq.* (6.6)*. Additionally, we assume that the initial error* $\hat{e}(t_0) = \hat{x}(t_0) - \hat{x}_E(t_0)$ *lies within the region* Ω_{ρ^*}*. For the closed-loop system of Eq.* (6.1) *under the encrypted lower layer feedback controller, we can determine positive real numbers* $\epsilon_{error}, \epsilon_w$*, for which there exist* $\Delta, \Delta_E, \gamma_E$*, and* d*, that satisfy the following conditions:*

$$|\dot{x}_E(t)| \leq \gamma_E < \frac{\hat{\theta}\alpha_3(\epsilon_{error})}{2L''_E + \alpha_4(\alpha_1^{-1}(\rho^*)) + \alpha_5(\alpha_1^{-1}(\rho^*) + M\Delta)} \tag{6.21}$$

$$\mu = \alpha_3^{-1}\left[\frac{(2L''_E + \alpha_4(\alpha_1^{-1}(\rho^*)) + \alpha_5(\alpha_1^{-1}(\rho^*) + M\Delta))\gamma_E}{\hat{\theta}}\right] \tag{6.22}$$

$$-(1-\hat{\theta})\alpha_3(\mu) + L'_w\theta + L'_eM\Delta + L'_E\gamma_E\Delta_E + e_q \leq -\epsilon_w/\Delta \tag{6.23}$$

for some $\hat{\theta}$ *with* $0 < \hat{\theta} < 1$*. If* $(\hat{x}(t_0) - \hat{x}_E(t_0)) \in \Omega_{\rho^*}$*, then the deviation* $\hat{e}(t)$ *remains bounded in* Ω_{ρ^*} *under the encrypted stabilizing controller and the actual closed-loop state trajectory* x *is always bounded in* Ω_ρ*. Furthermore, given a sufficiently large time* T*, the deviation between*

the actual system of Eq. (6.1) *and the economically optimal trajectory is ultimately bounded by* $|e(t)| \leq \epsilon_{error}$ *for* $t \in [t_k, t_k + t')$.

Proof. We prove that the deviation between the actual system evolution and economically optimal set point trajectory under the lower layer encrypted feedback controller (i.e., $\hat{e}(t)$) is always bounded in Ω_{ρ^*} and, after a sufficiently large time $T < t'$, where t' is the operating period of the LEMPC from t_0 to $t_0 + t'$, the deviation is bounded in $B_{\epsilon_{error}}$. Also, based on the bound derived in Eq. (6.18), we can say $e(t)$ is also bounded in Ω_{ρ^*} as $\hat{e}(t) \in \Omega_{\rho^*}$.

We assume that, at sampling time $t_k \in [t_0, t_0 + t')$, $\hat{e}(t_k) \in \Omega_{\rho^*} \backslash B_\mu$. At t_0, the LEMPC recomputes a new optimal trajectory $x_E(t)$ for the encrypted lower feedback layer to track from t_0 to $t_0 + t'$. We define two sets $B_{\epsilon_{error}} = \{|e(t)| \leq \epsilon_{error}\}$ and $B_\mu = \{|e(t)| \leq \mu\}$, where μ is defined in Eq. (6.22) and $B_\mu \subset B_{\epsilon_{error}}$. If the deviation $\hat{e}(t)$ is bounded in the set $\Omega_{\rho^*} \backslash B_\mu$ and the conditions of Eq. (6.21) and Eq. (6.22) are met, the deviation will decrease along the closed-loop state trajectory, and after a sufficiently large time T, the deviation will converge to the set B_μ. Furthermore, the deviation $e(t)$ is ultimately bounded in the ball $B_{\epsilon_{error}}$.

The time derivative of the control Lyapunov function along the deviation of system trajectory of Eq. (6.3) is, without disturbances or encryption:

$$\dot{V}(e(t_k), x(t_k)) = \frac{\partial V(e(t_k), x_E(t_k))}{\partial e} \dot{e}(t_k) + \frac{\partial V(e(t_k), x_E(t_k))}{\partial x_E} \dot{x}_E(t_k) \tag{6.24}$$

Using the Lipschitz property of Eq. (6.6b), after substituting $\dot{e}(t_k) = \dot{x}(t_k) - \dot{x}_E(t_k)$, we get

$$\begin{aligned}\dot{V}(e(t_k), x(t_k)) &= \frac{\partial V(e(t_k), x_E(t_k))}{\partial e}\dot{x}(t_k) - \frac{\partial V(e(t_k), x_E(t_k))}{\partial e}\dot{x}_E(t_k) + \frac{\partial V(e(t_k), x_E(t_k))}{\partial x_E}\dot{x}_E(t_k) \\ &\leq \frac{\partial V(e(t_k), x_E(t_k))}{\partial e} g(e(t_k), x_E(t_k), 0, h(e(t_k), x_e(t_k)), 0) \\ &\quad - \frac{\partial V(e(t_k), x_E(t_k))}{\partial e}\dot{x}_E(t_k) + \frac{\partial V(e(t_k), x_E(t_k))}{\partial x_E}\dot{x}_E(t_k) \\ &\leq -\alpha_3(|e(t_k)|) - \frac{\partial V(e(t_k), x_E(t_k))}{\partial e}\dot{x}_E(t_k) + \frac{\partial V(e(t_k), x_E(t_k))}{\partial x_E}\dot{x}_E(t_k)\end{aligned} \tag{6.25}$$

The time derivative of the control Lyapunov function along the deviation and economically optimal state trajectory for $\tau \in [t_k, t_k + \Delta)$, under the encrypted feedback controller, with disturbances is

$$\dot{V}(\hat{e}(\tau), \hat{x}(\tau)) = \frac{\partial V(\hat{e}(\tau), \hat{x}_E(\tau))}{\partial e}\dot{e}(\tau) + \frac{\partial V(\hat{e}(\tau), \hat{x}_E(\tau))}{\partial x_E}\dot{x}_E(\tau) \tag{6.26}$$

Adding and subtracting Eq. (6.24) to and from Eq. (6.26), we get

$$\begin{aligned}\dot{V}(\hat{e}(\tau), \hat{x}(\tau)) \leq &\frac{\partial V(\hat{e}(\tau), \hat{x}_E(\tau))}{\partial e}\dot{e}(\tau) - \frac{\partial V(e(t_k), x_E(t_k))}{\partial e}\dot{e}(t_k) + \frac{\partial V(e(t_k), x_E(t_k))}{\partial e}\dot{e}(t_k) \\ &+ \frac{\partial V(\hat{e}(\tau), \hat{x}_E(\tau))}{\partial x_E}\dot{x}_E(\tau) - \frac{\partial V(e(t_k), x_E(t_k))}{\partial x_E}\dot{x}_E(t_k)) \\ &+ \frac{\partial V(e(t_k), x_E(t_k))}{\partial x_E}\dot{x}_E(t_k))\end{aligned} \tag{6.27}$$

Substituting Eq. (6.25) in Eq. (6.27), using the bound of Eq. (6.2), and using the Lipschitz property

of Eq. (6.10), we get

$$\begin{aligned}
\dot{V}(\hat{e}(\tau),\hat{x}(\tau)) \leq & -\alpha_3(|e(t_k)|) - \frac{\partial V(e(t_k),x_E(t_k))}{\partial e}\dot{x}_E(t_k) + \frac{\partial V(e(t_k),x_E(t_k))}{\partial x_E}\dot{x}_E(t_k) \\
& - \frac{\partial V(e(t_k),x_E(t_k))}{\partial e}\dot{e}(t_k) - \frac{\partial V(e(t_k),x_E(t_k))}{\partial x_E}\dot{x}_E(t_k) \\
& + \frac{\partial V(\hat{e}(\tau),\hat{x}_E(\tau))}{\partial e}\dot{e}(\tau) + \frac{\partial V(\hat{e}(\tau),\hat{x}_E(\tau))}{\partial x_E}\dot{x}_E(\tau) \\
\leq & -\alpha_3(|e(t_k)|) + \frac{\partial V(\hat{e}(\tau),\hat{x}_E(\tau))}{\partial e}\dot{e}(\tau) - \frac{\partial V(e(t_k),x_E(t_k))}{\partial e}\dot{e}(t_k) \\
& + \frac{\partial V(\hat{e}(\tau),\hat{x}_E(\tau))}{\partial x_E}\dot{x}_E(\tau) - \frac{\partial V(e(t_k),x_E(t_k))}{\partial e}\dot{x}_E(t_k) \\
\leq & -\alpha_3(|e(t_k)|) + L'_w|w(\tau)| + L'_e|\hat{e}(\tau) - e(t_k)| + L'_E|\hat{x}_E(\tau) - x_E(t_k)| \\
& + L''_E|\dot{x}_E(\tau) - \dot{x}_E(t_k)| + L'_u|\hat{u} - u| + \alpha_5(|\hat{e}(\tau)|)\gamma_E + \alpha_4(|e(t_k)|)\gamma_E
\end{aligned} \tag{6.28}$$

Using the quantization error bounds of Eq. (6.17) and Eq. (6.18), in Eq. (6.28), we get

$$\begin{aligned}
\dot{V}(\hat{e}(\tau),\hat{x}(\tau)) \leq & -\alpha_3(|e(t_k)|) + L'_w|w(\tau)| + L'_e|\hat{e}(\tau) - e(\tau)| + L'_e|e(\tau) - e(t_k)| \\
& + L'_E|\hat{x}_E(\tau) - x_E(\tau)| + L'_E|x_E(\tau) - x_E(t_k)| \\
& + 2L''_E\gamma_E + L'_u 2^{-d-1} + \alpha_5(|\hat{e}(\tau)|)\gamma_E + \alpha_4(|e(t_k)|)\gamma_E \\
\leq & -\alpha_3(|e(t_k)|) + L'_w|w(\tau)| + L'_e 2^{-d} + L'_E 2^{-d-1} \\
& + L'_e|e(\tau) - e(t_k)| + L'_E|x_E(\tau) - x_E(t_k)| \\
& + 2L''_E\gamma_E + L'_u 2^{-d-1} + \alpha_5(|\hat{e}(\tau)|)\gamma_E + \alpha_4(|e(t_k)|)\gamma_E
\end{aligned} \tag{6.29}$$

Due to the continuity of $e(t)$ and $x_E(t)$ $\forall\, t \in [t_k, t_k + \Delta)$, and from Eq. (6.8), we can write that $|e(\tau) - e(t_k)| \leq M\Delta$, and $|x_E(\tau) - x_E(t_k)| \leq \gamma_E\Delta_E$ $\forall t \in [t_k, t_k + \Delta)$. Using these bounds, and

the inequalities of Eq. (6.6), it follows from Eq. (6.29):

$$\begin{aligned}\dot{V}(\hat{e}(\tau),\hat{x}(\tau)) \leq -\alpha_3(|e(t_k)|) + L'_w|w(\tau)| + L'_e 2^{-d} + L'_E 2^{-d-1} + 2L''_E\gamma_E + L'_u 2^{-d-1} \\ + L'_E\gamma_E\Delta_E + L'_e M\Delta + \alpha_5(|\hat{e}(\tau)|)\gamma_E + \alpha_4(|e(t_k)|)\gamma_E\end{aligned} \tag{6.30}$$

As $e(t_k) \in \Omega_{\rho^*}\backslash B_\mu$, Eq. (6.30) can be written as,

$$\begin{aligned}\dot{V}(\hat{e}(\tau),\hat{x}(\tau)) \leq -\alpha_3(\mu) + L'_w\theta + L'_e M\Delta + L'_E\gamma_E\Delta_E + e_q \\ + (\alpha_4(\alpha_1^{-1}(\rho^*)) + \alpha_5(\alpha_1^{-1}(\rho^*) + M\Delta) + 2L''_E)\gamma_E\end{aligned} \tag{6.31}$$

with $e_q = L'_e 2^{-d} + L'_E 2^{-d-1} + L'_u 2^{-d-1}$ representing the error due to quantization (for performing encryption). If Eq. (6.21) is satisfied, then there exists a γ_E such that the following equation holds:

$$\dot{V}(\hat{e}(\tau),\hat{x}(\tau)) \leq -(1-\hat{\theta})\alpha_3(\mu) + L'_w\theta + L'_e M\Delta + L'_E\gamma_E\Delta_E + e_q \tag{6.32}$$

for some positive $\hat{\theta} < 1$. If the condition of Eq. (6.23) is satisfied, then there exists $\epsilon_w > 0$ such that the following inequality holds for $\hat{e}(t_k) \in \Omega_{\rho^*}\backslash B_\mu$:

$$\dot{V}(\hat{e}(\tau),\hat{x}(\tau)) \leq -\epsilon_w/\Delta,\ \forall\tau \in [t, t_{k+1}) \tag{6.33}$$

Integrating this bound over $t \in [t_k, t_{k+1})$, we get

$$V(\hat{e}(t_{k+1}),\hat{x}(t_{k+1})) \leq V(\hat{e}(t_k),\hat{x}(t_k)) - \epsilon_w,\ \forall t \in [t_k, t_{k+1}) \tag{6.34}$$

$\forall\hat{e}(t_k) \in \Omega_{\rho^*}\backslash B_\mu$. Using the above inequalities recursively, if $e(t_k) \in \Omega_{\rho^*}\backslash B_\mu$, the deviation between the actual state trajectory and the economically optimal reference trajectory will converge

to B_μ, within time T, without exiting the set Ω_{ρ^*}. Further, there exists a sufficiently large $\epsilon_{error} > 0$, such that if the deviation exits the ball B_μ, it is still maintained within $B_{\epsilon_{error}}$ as the increase in deviation would be bounded over one sampling period. From the Lyapunov constraints of the LEMPC in Eq. (6.19f), and Eq. (6.19g), the reference trajectory $x_E(t)$ will be bounded in $\Omega_{\rho_{secure}}$ within time T. As $e(t)$ is always bounded in the set Ω_{ρ^*}, from Theorem 6.1, and $x(t) = x_E(t) + e(t)$, the closed-loop state trajectory of the system will converge to the set Ω_{ρ_e} in time T, where $\Omega_\rho < \Omega_{\rho_e} < \Omega_{\rho_{secure}}$, and will remain there. □

Remark 6.11. *From Eq.* (6.31)*, we can identify five factors affecting the rate of change of the control Lyapunov function when $\hat{e}(t_k) \in \Omega_{\rho^*} \backslash B_\mu$: the lower layer control system and LEMPC sampling periods (Δ and Δ_E), disturbance bound (θ), rate of change of the reference state trajectory ($\dot{x}_E$), and the quantization parameter (d). While disturbance is inherent to the system, adjustments to the other factors can be made to restrict the deviation between the state trajectory and reference state trajectory, thus achieving the desired tracking performance. In essence, decreasing the sampling times and the rate of change of the reference state trajectory while increasing the quantization parameter can help reduce the deviation between the actual state trajectories and reference trajectories.*

6.4 Application to a nonlinear chemical process

In this section, we apply the proposed encrypted two-layer control framework on a nonlinear chemical process with disturbance and sensor noise, operating at an unstable steady state. Multiple simulation cases are presented and compared to demonstrate the economic benefits and cyber-resilience of the proposed control framework.

6.4.1 Process description and model development

Specifically, the process considered is the conversion of reactant A to product B in a non-isothermal, well-mixed continuous stirred tank reactor (CSTR). This involves an irreversible second-order exothermic reaction, denoted as $A \rightarrow B$, with a reaction rate given by $r_B = k_0 e^{-\frac{E}{RT}} C_A^2$. The CSTR is equipped with a heating jacket that can either supply or remove heat at a rate Q. Using material and energy balance equations, we define the dynamic model of this process as follows:

$$\frac{\mathrm{d}C_A}{\mathrm{d}t} = \frac{F}{V}(C_{A0} - C_A) - k_0 \mathrm{e}^{\frac{-E}{RT}} C_A^2 \tag{6.35a}$$

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 \mathrm{e}^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{6.35b}$$

The reactor holds the reacting liquid with a constant volume V, C_A denotes the concentration of reactant A, and T represents the reactor temperature. The reactant A is introduced by the feed with a volumetric flow rate F, concentration C_{A0}, and a temperature of T_0. The liquid in the reactor maintains a constant heat capacity C_p and density ρ_L. Parameters such as ΔH, k_0, R, and E correspond to the enthalpy of reaction, pre-exponential constant, ideal gas constant, and activation energy, respectively. These parameters are quantified in Table 6.1. The state variables, expressed in deviation terms, consist of the reactant concentration and the reactor temperature, denoted as $x^\top = [C_A - C_{As},\ T - T_s]$, where the subscript "s" denotes the steady-state value. Initially, the CSTR operates at an unstable steady-state characterized by $[C_{As}, T_s] = [1.9537\ \mathrm{kmol/m^3}, 401.87\ \mathrm{K}]$, with inlet feed concentration and heat input rate denoted as $[C_{A0s},\ Q_s] = [4\ \mathrm{kmol/m^3}, 0\ \mathrm{kJ/hr}]$. The control inputs are: $C_{A0} - C_{A0_s}$ and $Q - Q_s$, representing deviations from the steady-state inlet concentration and heat input rate, respectively. These inputs are constrained within the closed

sets $[-3.5, 3.5]$ kmol/m^3 and $[-5 \times 10^5, 5 \times 10^5]$ kJ/hr, respectively. At the initial time $t = t_0$, the system begins at equilibrium ($x_0 = [0, 0]^\top$). Process noise, w_k, is introduced to the inlet flow rate, F, such that $|w_k| \leq 0.1 \times F$. Here, k denotes the sampling period, and w_k is a normally distributed random variable with zero mean and a standard deviation of 3.5% of the inlet flow rate of $5\,\mathrm{m}^3/\mathrm{hr}$. Additionally, non-Gaussian measurement noise, extracted from industrial data as described in [57], is added to all measured states. This noise is normalized and scaled by 1% before being applied to the concentration state, while it is applied to the temperature state without scaling. The control objective is to increase the economic profit of the process described in Eq. (6.35) by

Table 6.1: Parameter values for the chemical process example

$F = 5\,\mathrm{m}^3/\mathrm{hr}$	$V = 1\,\mathrm{m}^3$
$k_0 = 8.46 \times 10^6\,\mathrm{m}^3/(\mathrm{kmol\,hr})$	$E = 5 \times 10^4\,\mathrm{kJ/kmol}$
$R = 8.314\,\mathrm{kJ}/(\mathrm{kmol\,K})$	$\rho_L = 1000\,\mathrm{kg/m}^3$
$\Delta H = -1.15 \times 10^4\,\mathrm{kJ/kmol}$	$T_0 = 300\,\mathrm{K}$
$Q_s = 0\,\mathrm{kJ/hr}$	$C_{A0_s} = 4\,\mathrm{kmol/m}^3$
$C_{As} = 1.9537\,\mathrm{kmol/m}^3$	$T_s = 401.87\,\mathrm{K}$
$C_p = 0.231\,\mathrm{kJ}/(\mathrm{kg\,K})$	

manipulating the inlet concentration and heat input rate, while ensuring that the state trajectories of the closed-loop system remains within the stability region Ω_ρ at all times using the LEMPC. Ultimately, the system should converge to the economically viable region Ω_{ρ_e} and stay there. The objective function of the LEMPC optimizes the production rate of B, consumption of reactant A, and the heat input rate $Q - Q_s$ as follows:

$$L(x_E, u) = A_1 k_0 e^{-\frac{E}{RT}} C_A^2 - A_2(C_{A0} - C_{A0_s}) - A_3(Q - Q_s)^2 \tag{6.36}$$

where A_1, A_2, and A_3 are the potentially time-varying weighting factors that account for fluctuations in process economics, i.e. product selling price, reactant cost, and energy cost, respectively. The control Lyapunov function $V(e, x_E) = x_E^\top P x_E$ is defined with the following positive definite P matrix:

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{6.37}$$

The time-varying weights chosen for the example considered are provided in Table 6.2. The closed-

Table 6.2: Time-varying LEMPC weights for chemical process example

Time (t)	A_1	A_2	A_3
$0\,\text{hr} \leq t < 1\,\text{hr}$	1	17	1×10^{-8}
$1\,\text{hr} \leq t < 2\,\text{hr}$	0.99	14	0.8×10^{-8}
$2\,\text{hr} \leq t < 3\,\text{hr}$	1.01	5	0.84×10^{-8}
$3\,\text{hr} \leq t < 4\,\text{hr}$	0.98	7	0.9×10^{-8}
$t \geq 4\,\text{hr}$	1.02	9	0.9×10^{-8}

loop stability region for the CSTR is defined as Ω_ρ, with $\rho = 320$, which is characterized as a level set of the Lyapunov function. The secure operating region $\Omega_{\rho_{secure}}$ for the LEMPC described in Eq. (6.19) is defined with $\rho_{secure} = 85$. Further, the desired region of economic feasibility, Ω_{ρ_e}, within which the real state trajectory is to be bounded, is selected to have ρ_e = 130. The operating period of the LEMPC is $t' = 1\,\text{hr}$. The lower layer encrypted control system operates with a sampling period of $1.8\,\text{s}$, whereas the LEMPC has a sampling period of $180\,\text{s}$. The prediction horizon for the LEMPC is set to $N_E = 20$ sampling periods. The integration step h_c chosen to integrate the LEMPC objective function using the explicit Euler method is $0.36\,\text{s}$. The positive definite matrix P in $V = x_E^\top P x_E$ and the stability region Ω_ρ are determined through simulations that search for the largest invariant set Ω_ρ in the state-space within which $\dot{V}$ is rendered negative,

for all states in Ω_ρ under the stabilizing controller $h(e, x_E) \in U$. In the present example, $h(e, x_E)$ is a set of PI controllers, $[u_1, u_2]^\top$ of the form of Eq. (6.20) with proportional gains $K_1 = 10^1$ and $K_2 = 10^4$, and integral time constants $\tau_1 = 10^{-3}$ and $\tau_2 = 10^{-6}$.

6.4.2 Performing encryption in the two-layer control framework

Before encrypting and decrypting the data, parameters such as d, l_1, and l_2 are carefully selected. The integer bit count $l_1 - d$ is determined from extreme feasible states and control inputs. The upper limit of $\mathbb{Q}_{l_1,d}$ is calculated using from $2^{l_1-d-1} - 2^{-d}$, while the lower limit can be obtained from -2^{l_1-d-1}. The quantization parameter, d, is selected depending on the desired level of accuracy and operating range of state and control input values. Further, l_2 is chosen to exceed l_1. In the example presented in this section, $l_1 - d$ is calculated to be 16, determining l_1 and d. In the set $\mathbb{Q}_{l_1,d}$, numbers are separated by a resolution of 2^{-d}. In our simulations, we use $d = 8$ in all scenarios except when it is specifically changed and noted to be $d = 1$. For $d = 8$, $l_1 = 24$, and l_2 is selected as 30. Similarly, for $d = 1$, $l_1 = 16$, and l_2 is set to 20. Paillier Encryption is implemented using Python's "phe" module, PythonPaillier [21]. To solve the multi-constrained non-convex optimization problem of the upper layer LEMPC in the two-layer encrypted control framework, we utilize the Python module of the IPOPT software [83].

6.4.3 Cyberattack detection and system reconfiguration

A logic-based cyberattack detector is integrated into the lower layer of the encrypted two-layer control framework. This detector receives sensor readings every three sampling instances of the lower control layer and utilizes this data to compute the control Lyapunov function $V(x)$. Importantly,

this computation occurs prior to encryption or transmission to the cloud HSM, ensuring its security. In the event of a cyberattack, the objective of the attack is to divert the process from its operating trajectory while still maintaining it within the stability region, Ω_ρ. This may lead to a prolonged cyberattack that could go undetected, potentially being mistaken for a process disturbance. The upper layer LEMPC aims to maintain the set point trajectory within a more conservative region, $\Omega_{\rho_{secure}}$, and lacks information about the bounded region Ω_{ρ_e} as detailed in the earlier section. In the event of an attack, the system would drift away from the economically viable operating region, making it challenging to bring it back within $\Omega_{\rho_{secure}}$ due to the attack's interference. If the system were not under attack, the contractive constraints of the LEMPC would drive the system back toward the secure operating region. If the detector records three consecutive instances where the control Lyapunov function has values $V(x) \geq \rho_e$, and its value increases compared to its last recorded value, it identifies the system as being under attack. Subsequently, the control reconfiguration logic rejects the previously received economically optimal set points from the compromised upper layer. Subsequently, it utilizes the encrypted set points of the prior operating period when the system operated without attack detection.

Remark 6.12. *In the proposed control architecture, the lower-layer control system receives encrypted set-points (values of the operating trajectory at the current time) that are maintained at different time intervals. The control actions implemented on the process by the lower-layer control system are calculated from encrypted feedback without decrypting the state information or the set points. Since the measured state data remains encrypted, it is very difficult to implement a cyberattack in the lower-layer control system; this is an important advantage of the proposed control architecture. On the other hand, cyberattacks can be launched in the upper-layer EMPC system that calculates the set-points for the lower-layer control system and this is where attack detection mechanisms are implemented to detect such attacks. With respect to cyberattacks that can influ-*

ence encrypted communication, this is an issue that goes beyond the scope of the present work. It is important to note that given the linear nature of the lower-layer control system, alternative, perhaps more secure, encryption schemes can be used in the lower-layer with similar properties being proved for the closed-loop system.

Remark 6.13. *Since the lower layer solely receives encrypted set points from the upper layer and operates within the defined economically viable region, taking into account fluctuating economics, identifying cyberattacks that do not push the system outside this region becomes challenging. However, the absence of information in the compromised upper layer concerning the bounds of this region adds another layer of robustness to the proposed detection scheme. Detecting attacks within this region would necessitate decrypted economic information from the upper layer, which could also be vulnerable to cyberattacks. Therefore, in this study, we only focus on cyberattacks capable of driving the system away from the economically viable operating region while maintaining it within the stable region.*

Remark 6.14. *As mentioned in Section* 6.3.1*, there are three potential points of cyberattack where plaintext data could be manipulated in the presented closed-loop design of Figure* 6.1*. For brevity, we only demonstrate results when a false-data injection cyberattack is initiated on the data received by the upper layer LEMPC after decryption, ensuring that the system does not exit the stability region. Detailed information on the launched cyberattack and similar classes of cyberattacks has been discussed in [41]. These attacks are designed to ensure that the system does not exit the stability region during the attack, making them difficult to detect.*

6.4.4 Simulation results of the encrypted two-layer control framework

The proposed encrypted two-layer control framework is applied to a nonlinear chemical process with sensor noise and disturbances. Results depicted in Figures 6.2 and 6.3 illustrate the proposed two-layer framework's performance under an LEMPC objective function whose coefficients (weights) change for each operating period.Figures 6.4 and 6.5 display the closed-loop states and inputs and corresponding state-space trajectories under encrypted lower-layer control with set-points calculated. Furthermore, Figures 6.6 and 6.7 show the results under the encrypted two-layer control framework with an LEMPC objective function whose coefficients are set equal to the ones of the first operating period throughout the five-period operation. Finally, Figures 6.8 and 6.9 illustrate closed-loop states, inputs and state-space trajectories under encrypted lower-layer control with set-points calculated at the upper layer using steady-state optimization with the same economic objective as in the LEMPC and with weights set equal to the ones of the first operating period.

Analyzing these results in more detail, the closed-loop simulation results in Figures 6.2 and 6.3 illustrate time-varying operating trajectories for different operating periods. Initially, when raw material costs are high, a time-varying operation is preferred to maximize economic benefits. As raw material costs decrease over successive periods, steady-state operation becomes more favorable as determined by the upper-layer LEMPC. Figures 6.4 and 6.5 depict how different steady-states are maintained for each operating period with time-varying (changing every period and remaining constant within a single period) weights in the objective function of the steady-state optimizer. Figures 6.6 and 6.7 show that with time-invariant weights in the LEMPC objective function,

similar dynamic trajectories are computed and maintained across operating periods. Figures 6.8 and 6.9 demonstrate that steady-state operation is maintained when a time-invariant objective is used in the steady-sate optimizer for all operating periods. Comparing the performance of these scenarios justifies the application of EMPC through the encrypted two-layer framework to achieve economically optimal time-varying operation in certain periods over steady-state operation.

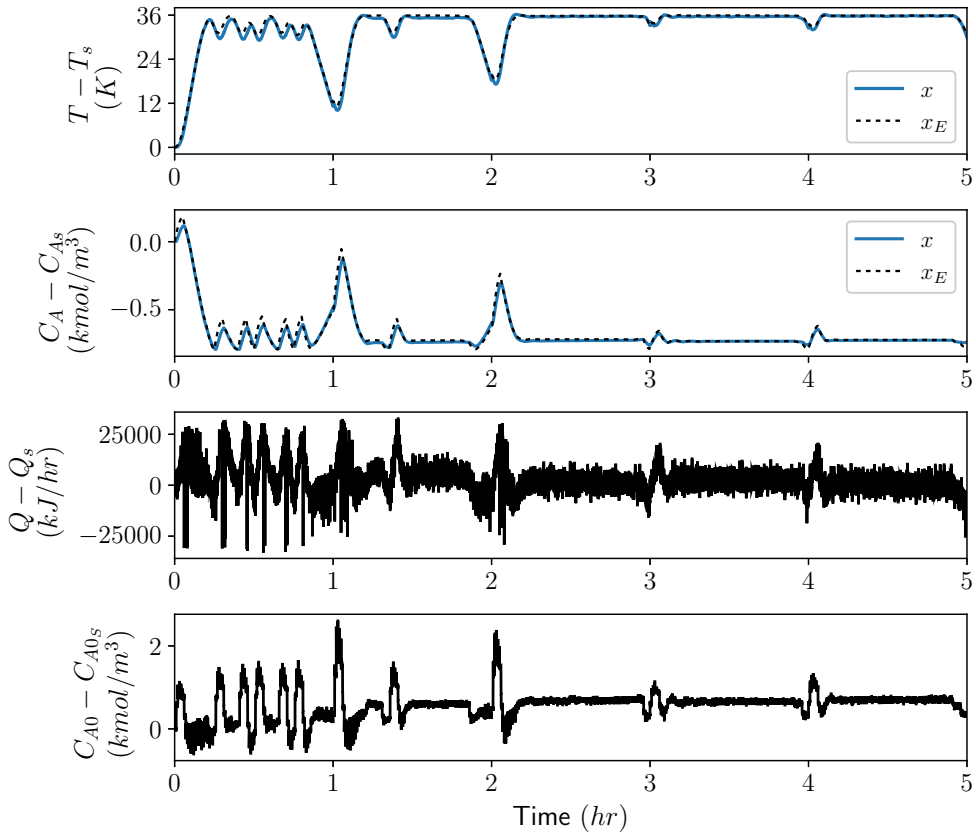

Figure 6.2: State and control input profiles under the encrypted two-layer control framework with an LEMPC objective function whose weights change for each operating period.

Table 6.3 presents the total economic objective function values for the closed-loop simulations. These results demonstrate that the proposed framework, particularly with dynamic economic optimization, outperforms steady-state optimizers. Notably, the time-varying LEMPC objective function yields the highest economic objective function after $5\,\mathrm{hr}$ of process time, followed by the

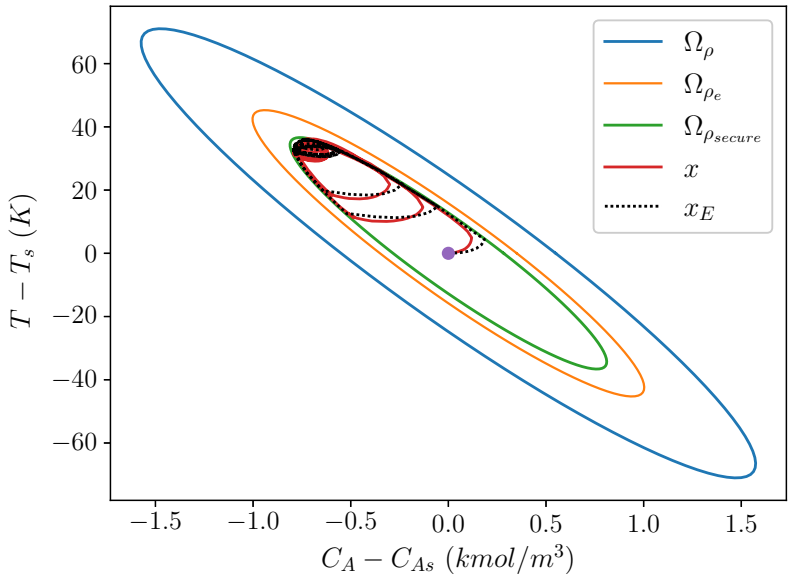

Figure 6.3: State-space plot for the evolution of the state and reference trajectories under the encrypted two-layer control framework with an LEMPC objective function whose weights change for each operating period.

Table 6.3: Economic Objective function values for different simulations at the end of a $5\,\mathrm{hr}$ process duration

Operation type			
Objective function weights	Optimization	Total economic objective function	Increase (%)
Time-varying	LEMPC	70,569	47.66
	Steady-state	56,541	18.30
Time-invariant	LEMPC	65,614	37.28
	Steady-state	47,793	0

LEMPC with a constant objective function. In all aforementioned cases, the lower layer encrypted feedback controllers track the state trajectory well, and it remains bounded in Ω_{ρ_e} at all times. Figure 6.11 and Figure 6.10 depict results under the encrypted two-layer control framework with a time-varying LEMPC objective function, with and without a cyberattack detection and reconfiguration mechanism, respectively. In both cases, a false-data injection attack is initiated at $4\,\mathrm{hr}$. With the detection and reconfiguration mechanism, the state trajectory is promptly tracked within

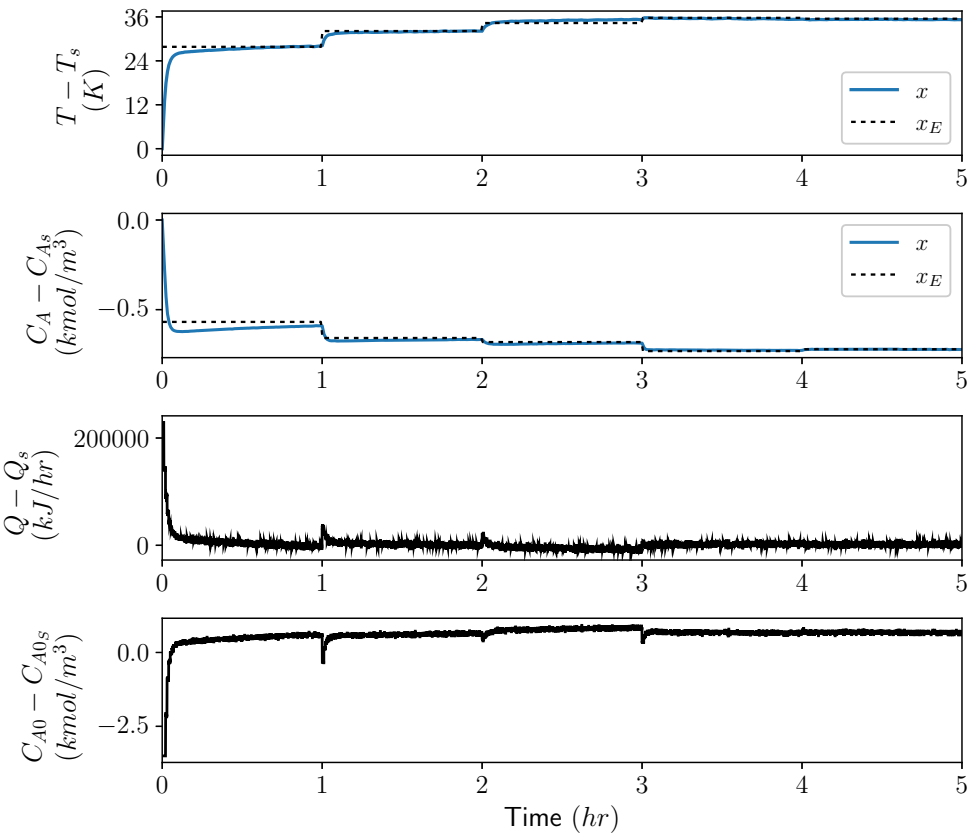

Figure 6.4: State and control input profiles under encrypted lower-layer control with set-points calculated at the upper layer using steady-state optimization with the same economic objective as in the LEMPC and with weights changing at each operating period.

Ω_{ρ_e} upon exit as shown in Figure 6.11, while without it, the trajectory remains outside Ω_{ρ_e} for an extended period as depicted in Figure 6.10. After detection, the lower-layer controller follows the state trajectory from the previous operating period, during which no attack was detected, and the closed-loop state remained within Ω_{ρ_e} at all times. In all these simulations, the quantization parameter d is maintained at 8. Figure 6.12 illustrates results under the encrypted two-layer control framework with a time-varying LEMPC objective function for $d = 1$, where the state trajectory exits Ω_{ρ_e} at certain points and struggles to track the set point trajectory effectively, unlike the other cases. All other parameters were maintained the same as the case presented in Figure 6.3, for

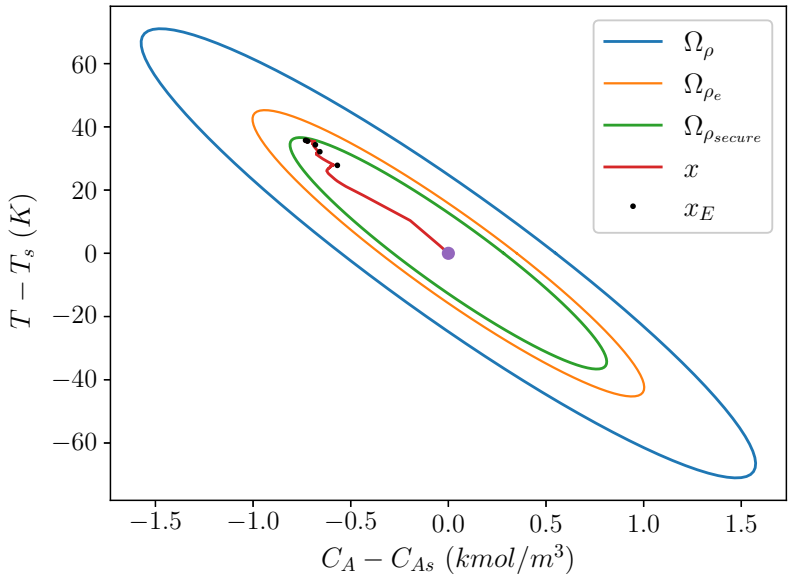

Figure 6.5: State-space plot for the evolution of the state and reference trajectories under encrypted lower-layer control with set-points calculated at the upper layer using steady-state optimization with the same economic objective as in the LEMPC and with weights changing at each operating period.

comparison. This highlights the need for using a higher quantization parameter and validates the theoretical results.

Remark 6.15. *As previously mentioned, we have employed both time-varying and time-invariant coefficients (weights) in the objective functions across different scenarios. However, the coefficients remain the same for the initial operating period in all cases. In conducting the economic performance comparison presented in Table* 6.3*, we utilized recorded state and control input trajectory data from the different closed-loop simulations spanning a process duration of 5 hours. This data was used to compute the total objective function value with time-varying (changing from period to period and staying constant within a single period) coefficients in all listed scenarios. This approach facilitates a quantitative comparison of the potential loss or gain resulting from the utilization or omission of time-varying coefficients in the objective function.*

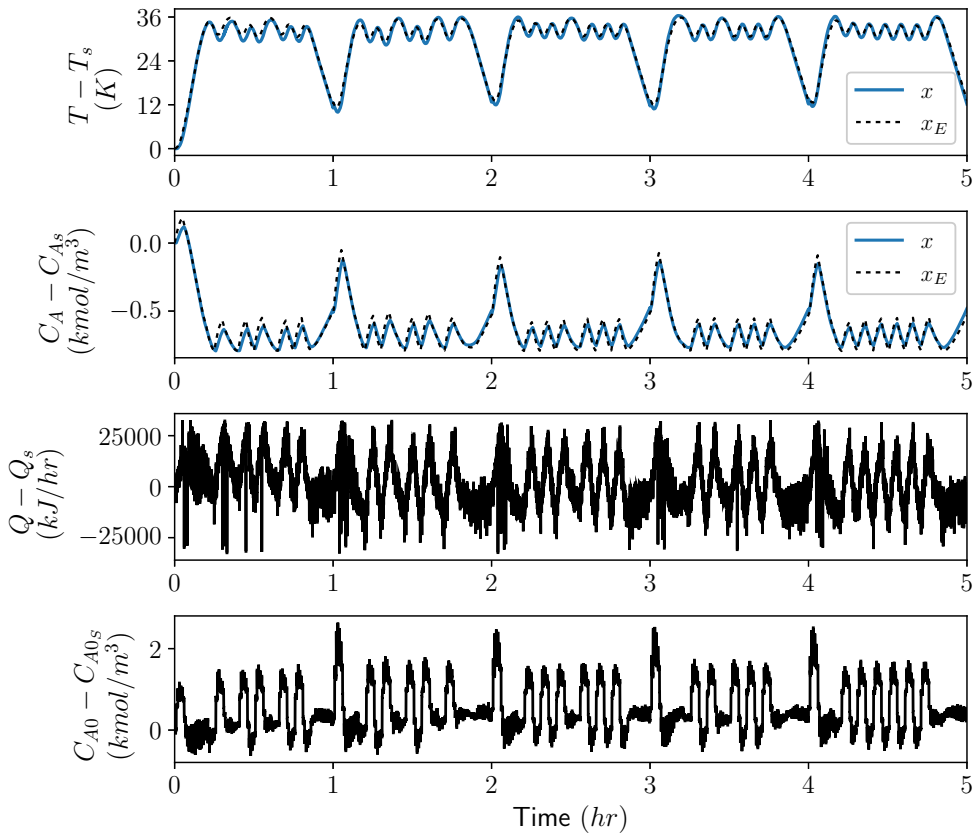

Figure 6.6: State and control input profiles under the encrypted two-layer control framework with an LEMPC objective function that uses the same weights for each operating period.

Remark 6.16. *In Figure* 6.12*, the actual state trajectory exits the economically optimal operating region. This is attributed to the use of a different value for the quantization parameter,* $d = 1$*, as opposed to* $d = 8$*, resulting in a different bounded error and consequently, a distinct economically optimal region* Ω_{ρ_e}*. Despite this variation, we have depicted the same regions for comparison purposes, emphasizing that opting for a higher quantization parameter enables a stricter bounded error. Similarly, maintaining lower layer sampling times, and a lower rate of change of the state reference trajectory can lead to stricter bounds on the error.*

Remark 6.17. *With respect to comparing the proposed cybersecure, two-layer control architecture to other approaches, it is important to point out that it is not as optimal as the use of a single-*

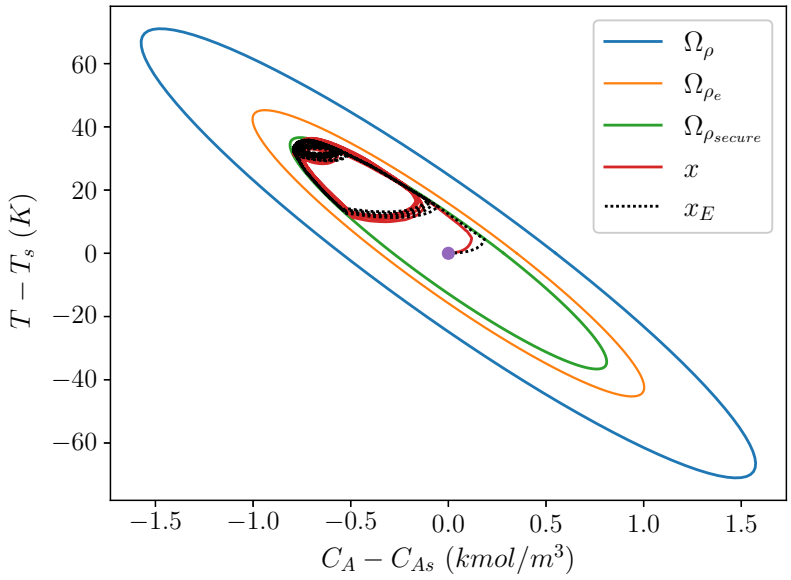

Figure 6.7: State-space plot for the evolution of the state and reference trajectories under the encrypted two-layer control framework with an LEMPC objective function that uses the same weights for each operating period.

layer EMPC system where there is no need to impose rate of change constraints in the operating trajectory calculated by the EMPC. However, such a single-layer EMPC system (in addition to requiring a significant computational load at the lower layer) is fully non-robust to cyber-attacks as it requires decrypted signals to carry out calculations in the feedback control layer, rendering it vulnerable to cyber-attacks. If, on the other hand, one were to compare the two-layer control architecture with encryption at the lower layer to the same architecture without encryption, then the performance loss is relatively small when a sufficiently large d value is used as we have demonstrated above (see also [81]).

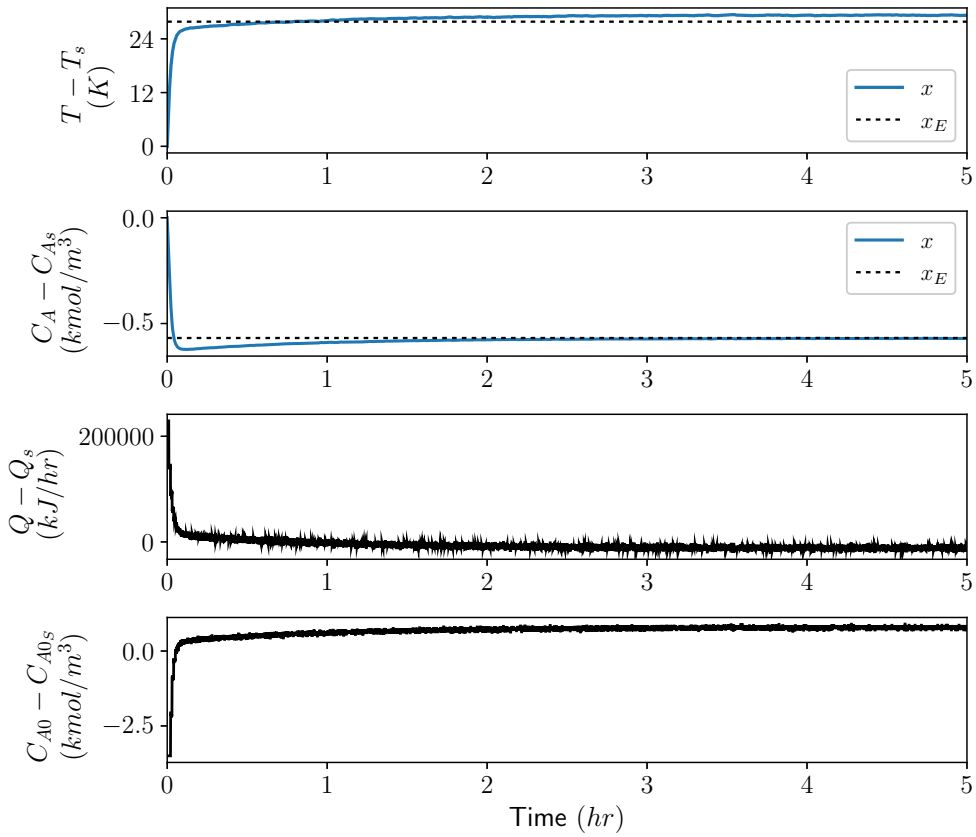

Figure 6.8: State and control input profiles under encrypted lower-layer control with set-points calculated at the upper layer using steady-state optimization with the same economic objective as in the LEMPC and with fixed weigths.

6.5 Conclusion

In this chapter, we introduced an encrypted two-layer framework to integrate dynamic economic optimization with encrypted control for nonlinear processes. At the upper layer, an LEMPC with a time-varying objective function computed the economically optimal state trajectories to be tracked by the encrypted lower layer feedback control system. Through a comprehensive stability analysis, we established bounds on the deviation between the actual state trajectory and reference trajectory, and listed tunable parameters to achieve the desired bounded error. Theoretical results were demonstrated and validated using a chemical process example, and the economic benefits of the

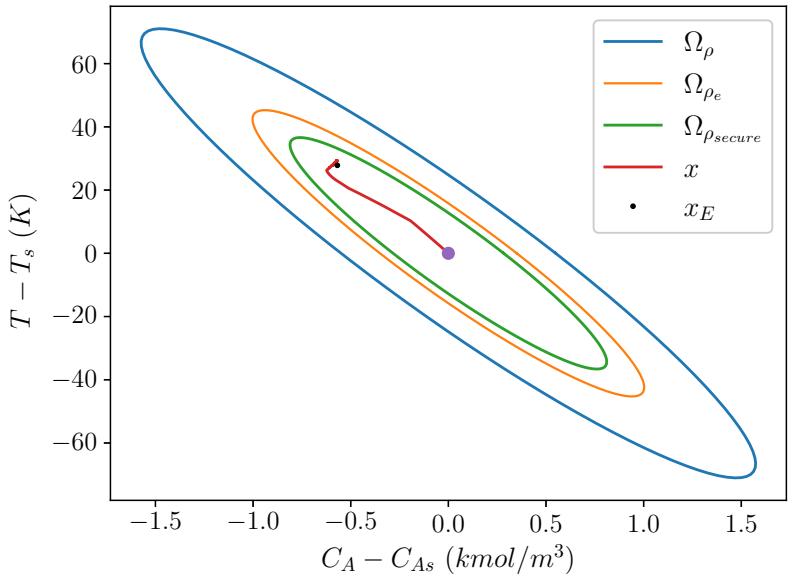

Figure 6.9: State-space plot for the evolution of the state and reference trajectories under encrypted lower-layer control with set-points calculated at the upper layer using steady-state optimization with the same economic objective as in the LEMPC and with fixed weights.

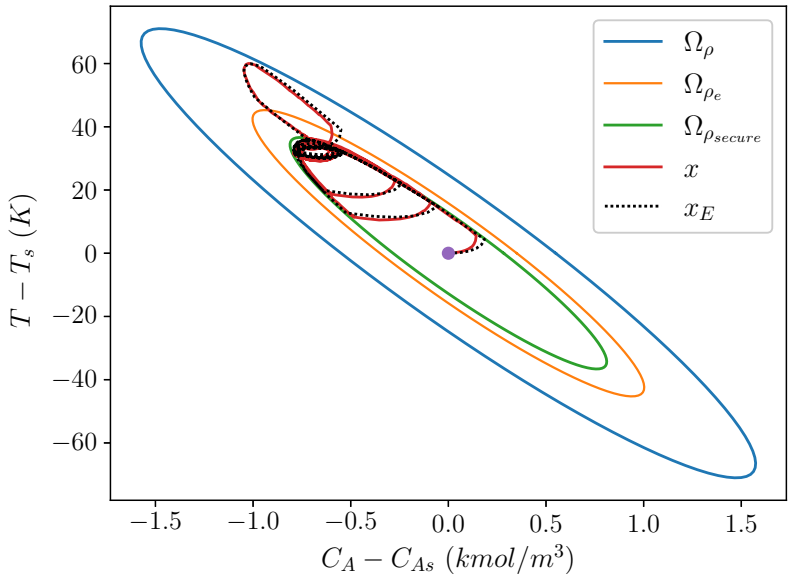

Figure 6.10: State-space plot for the evolution of the state and reference trajectories under the encrypted two-layer control framework using an LEMPC objective function whose weights change for each operating period, without cyberattack detection and reconfiguration, when a cyberattack is initiated at $t = 4\,\mathrm{hr}$.

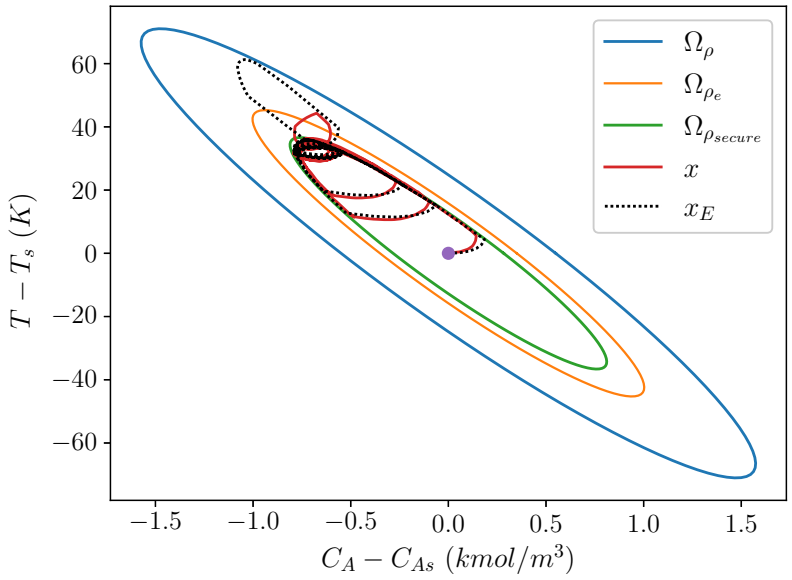

Figure 6.11: State-space plot for the evolution of the state and reference trajectories under the encrypted two-layer control framework using an LEMPC objective function whose weights change for each operating period, with cyberattack detection and reconfiguration, when a cyberattack is initiated at $t = 4\,\text{hr}$.

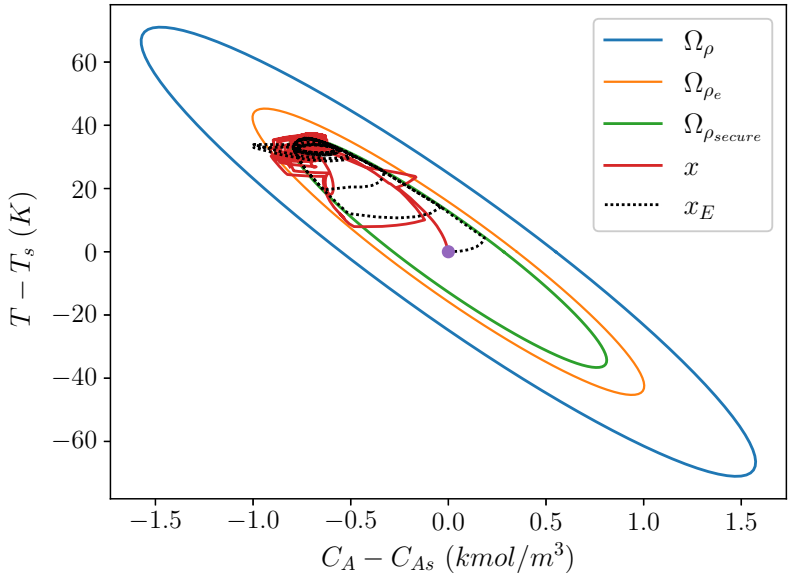

Figure 6.12: State-space plot for the evolution of the state and reference trajectories under the encrypted two-layer control framework using an LEMPC objective function whose weights change for each operating period, with $d = 1$.

encrypted two-layer control framework were showcased. Moreover, we demonstrated the cyber-resilience of the proposed control framework through cyberattack detection and reconfiguration mechanisms when the system was subjected to a cyberattack.

Chapter 7

Conclusion

This thesis discusses a number of different designs of MPC systems with encrypted communication between different components of the control system to improve the confidentiality of data transmitted, cybersecurity, and ensure cyber-resilient operation of nonlinear chemical processes. Firstly, an encrypted centralized LMPC system is proposed, the effect of quantization on closed-loop performance is demonstrated, and its computational burden is studied. Next, a two-tier LMPC system with machine-learning-based cyberattack detection is proposed with attack-resilient operation strategies when decryption occurs in cyber-vulnerable environments. Next, an encrypted decentralized model predictive control scheme for nonlinear time-delay systems with rigorous theoretical analysis on their closed-loop stability properties. Furthermore, encrypted distributed model predictive control systems with extended Luenberger observer-based state estimations for nonlinear processes when only partial state measurements are available. Lastly, an encrypted two-layer control framework to maximize economic performance while addressing fluctuating real-world economic with cyberattack resilient operation.

In Chapter 2, we developed and applied an Encrypted Lyapunov-based model predictive control (LMPC) Scheme to a large-scale chemical process network involved in the production of ethylbenzene. By employing the encrypted LMPC, we conducted closed-loop simulations for different quantization parameters and identified errors resulting from quantization. We illustrated that the effect of quantization could be more profound than plant/model mismatch when a low quantization parameter is chosen. To mitigate the impact of quantization, we proposed using a higher quantization parameter, specifically $d = 8$. Furthermore, through a comprehensive analysis of the duration of encryption-decryption at each sampling instance, we observed that the computational burden on the control input calculation time remained consistent across all tested quantization parameters. This finding supports the recommendation of employing a higher quantization parameter, as it not only minimizes the impact of quantization errors but also ensures secure communication between the sensor-controller and controller-actuator, thus enhancing system cybersecurity without compromising the performance of the controller.

In Chapter 3, we presented an encrypted two-tier control architecture incorporating an ML-based cyberattack detector to enhance the operational safety, cybersecurity, and closed-loop performance of nonlinear process systems. The lower-tier control system comprises a set of encrypted proportional-integral controllers, while the upper-tier control system employs an encrypted Lyapunov-based model predictive controller. This architecture enhances system cybersecurity, even in settings where control input computations may not be cybersecure. By integrating both linear and nonlinear controllers with encryption, the developed two-tier control architecture can be adapted to large-scale nonlinear processes. Further, we have provided insights into the framework and formulation of the encrypted lower- and upper-tier control systems. Through a comprehensive

stability analysis, we have identified potential sources of error and established bounds to ensure closed-loop system stability. Additionally, we have delved into the development of an ML-based cyberattack detector, addressed critical aspects such as quantization parameter selection, sampling time criteria, and computational load assessment. These issues are essential for the practical implementation of the proposed control system across nonlinear processes. To validate the efficacy of our control framework, we subjected it to previously unseen cyberattack patterns within a nonlinear chemical process network utilized in ethylbenzene production. We carried out a detailed simulation study that exposed the implementation and performance of the two-tier control architecture and the usefulness of the cyberattack detector.

In Chapter 4, we devised and applied an encrypted decentralized control architecture to a large-scale nonlinear chemical process network with input and state delays. A stability analysis of the encrypted decentralized MPC applied to a nonlinear system with state delays was conducted, yielding bounds on the errors due to quantization, state delays, and sample-and-hold implementation of the controller. Based on these bounds, the system can be stabilized within the desired stability region. We established guidelines to implement this control structure in any nonlinear process, such as selection of parameters l_1, l_2, and d for quantization, and the sampling time criterion. The encrypted decentralized LMPC employs a DDE model to account for state delays in the process. Closed-loop simulations are compared with and without the incorporation of a predictor into the LMPC design, where the predictor predicts the state values after the input delay period. A significant improvement in the closed-loop performance was observed with the integration of the predictor, as the states and inputs converged to their steady state values with negligible oscillations. Also, with the inclusion of the predictor, states converged within the desired stability region rep-

resented by the level set $\Omega_{\rho_{\min}}$. However, without the predictor, the states only stabilize within the larger level set Ω_ρ and with oscillations. Thus, by employing the encrypted decentralized LMPC with predictor feedback, we were able to reduce the computation time and complexity of the control problem, improve the closed-loop performance, and enhance the cybersecurity of the control system.

In Chapter 5.1, we introduced and applied encrypted distributed control architectures, both sequential and iterative, with state estimation, to a large-scale nonlinear chemical process network utilizing partial state feedback with sensor noise. We established practical guidelines for implementing this control structure in any nonlinear process by including the selection of key parameters such as l_1, l_2, and d for quantization, and the criterion for setting the sampling time. Through closed-loop simulations, we demonstrated that both the sequential and iterative distributed LMPCs, with encrypted communication between the sensor–controller and controller–actuator links, could stabilize the system within the desired stability region using the extended Luenberger observer for state estimation, in a finite process simulation time. Furthermore, we conducted a comprehensive comparative analysis of various encrypted control strategies, including centralized, decentralized, and distributed approaches with state estimation. The computational time, closed-loop performance, and suitability of the different encrypted control architectures were discussed. In conclusion, our findings indicate that the encrypted iterative distributed LMPC emerges as the most suitable choice for enhancing the cybersecurity of large and complex systems, with highly coupled dynamics between states. This approach reduces the computational complexity associated with centralized control, leverages controller communication to improve closed-loop performance, and maintains a reasonable computation time, while enhancing the cybersecurity

of the control system. Following this, in Chapter 5.2 we also formulated an encrypted iterative distributed LMPC system employing encrypted signals for data transmission between sensors, controllers, and actuators. Following a comprehensive stability analysis, we determined bounds for errors from quantization, process disturbances, and the sample-and-hold implementation of the controller. With these bounds, the system could be stabilized within the desired stability region. Selection of encryption-decryption key lengths, quantization parameters, sampling time criterion, and potential methods to decrease the encryption–decryption time were discussed to facilitate practical implementation. Closed-loop simulations were performed, comparing the proposed control scheme against the encrypted centralized LMPC. Non-Gaussian sensor noise obtained from an industrial data set and process disturbances were used to demonstrate the industrial relevance and suitability of the proposed approach. The results favor the use of the encrypted distributed LMPC system, which not only improves closed-loop performance but also significantly reduces the computational time needed to calculate the control input, positioning the encrypted iterative distributed LMPC as an effective solution for improving closed-loop performance, decreasing computational time, and enhancing cybersecurity in large-scale nonlinear systems.

In Chapter 6, we introduced an encrypted two-layer framework to integrate dynamic economic optimization with encrypted control for nonlinear processes. At the upper layer, an LEMPC with a time-varying objective function computed the economically optimal state trajectories to be tracked by the encrypted lower layer feedback control system. Through a comprehensive stability analysis, we established bounds on the deviation between the actual state trajectory and reference trajectory, and listed tunable parameters to achieve the desired bounded error. Theoretical results were demonstrated and validated using a chemical process example, and the economic benefits

of the encrypted two-layer control framework were showcased. Moreover, we demonstrated the cyber-resilience of the proposed control framework through cyberattack detection and reconfiguration mechanisms when the system was subjected to a cyberattack.

Bibliography

[1] Agrawal, S. Agrawal, J., 2015. Survey on anomaly detection using data mining techniques. Procedia Computer Science, 60, 708–713.

[2] Al-Abassi, A., Karimipour, H., Dehghantanha, A., Parizi, R. M., 2020. An ensemble deep learning-based cyber-attack detection in industrial control system. IEEE Access, 8, 83965–83973.

[3] Alhajeri, M. S., Wu, Z., Rincon, D., Albalawi, F., Christofides, P. D., 2021. Machine-learning-based state estimation and predictive control of nonlinear processes. Chemical Engineering Research and Design, 167, 268–280.

[4] Ali, J. M., Hoang, N. H., Hussain, M. A., Dochain, D., 2015. Review and classification of recent observers applied in chemical process systems. Computers & Chemical Engineering, 76, 27–41.

[5] Alnajdi, A., Suryavanshi, A., Alhajeri, M. S., Abdullah, F., Christofides, P. D., 2023. Machine learning-based predictive control of nonlinear time-delay systems: Closed-loop stability and input delay compensation. Digital Chemical Engineering, 7, 100084.

[6] Amrit, R., Rawlings, J. B., Angeli, D., 2011. Economic optimization using model predictive control with a terminal cost. Annual Reviews in Control, 35(2), 178–186.

[7] Bakule, L., 2008. Decentralized control: An overview. Annual reviews in control, 32(1), 87–98.

[8] Barker, E. Barker, W., 2019. Recommendation for key management, part 2: best practices for key management organization. National Institute of Standards and Technology.

[9] Barrett, M. P., 2018. Framework for improving critical infrastructure cybersecurity (version 1.1). NIST Cybersecurity Framework.

[10] Bemporad, A., Heemels, M., Johansson, M., 2010. Networked control systems, Volume 406. Springer.

[11] Bomze, I. M., Demyanov, V. F., Fletcher, R., Terlaky, T., 2010. Nonlinear optimization: lectures given at the CIME Summer School held in Cetraro, Italy, July 1-7, 2007. Springer.

[12] Carvalho, D., Morais, J., Almeida, J., Martins, P., Quental, C., Caldeira, F., 2019. A technical overview on the usage of cloud encryption services. In European Conference on Cyber Warfare and Security, 733–XI. Academic Conferences International Limited.

[13] Chen, S., Wu, Z., Christofides, P. D., 2020. A cyber-secure control-detector architecture for nonlinear processes. AIChE Journal, 66, e16907.

[14] Chen, S., Wu, Z., Christofides, P. D., 2020. Decentralized machine-learning-based predictive control of nonlinear processes. Chemical Engineering Research and Design, 162, 45–60.

[15] Christofides, P. D., Scattolini, R., Pena, de la D. M., Liu, J., 2013. Distributed model predictive control: A tutorial review and future research directions. Computers & Chemical Engineering, 51, 21–41.

[16] Conklin, W. A., 2016. IT vs. OT security: A time to consider a change in CIA to include resilienc. In Proceedings of 49th Hawaii International Conference on System Sciences, 2642–2647, Koloa, Hawaii.

[17] Damgård, I., Jurik, M., Nielsen, J. B., 2010. A generalization of paillier's public-key system with applications to electronic voting. International Journal of Information Security, 9, 371–385.

[18] Darup, M. S., 2020. Encrypted MPC based on ADMM real-time iterations. IFAC-PapersOnLine, 53, 3508–3514.

[19] Darup, M. S., Redder, A., Shames, I., Farokhi, F., Quevedo, D., 2017. Towards encrypted MPC for linear constrained systems. IEEE Control Systems Letters, 2, 195–200.

[20] Darup, M. S., Redder, A., Quevedo, D. E., 2018. Encrypted cloud-based MPC for linear systems with input constraints. IFAC-PapersOnLine, 51, 535–542.

[21] Data61, C. Python paillier library. https://github.com/data61/python-paillier, 2013. accessed January 10, 2024.

[22] Davies, R., 2015. Industry 4.0: Digitalisation for productivity and growth.

[23] Dochain, D., 2003. State and parameter estimation in chemical and biochemical processes: a tutorial. Journal of Process Control, 13, 801–818.

[24] Durand, H., 2018. A nonlinear systems framework for cyberattack prevention for chemical process control systems. Mathematics, 6, 169.

[25] Durand, H. Wegener, M., 2020. Mitigating safety concerns and profit/production losses for chemical process control systems under cyberattacks via design/control methods. Mathematics, 8, 499.

[26] Dutta, V., Choraś, M., Pawlicki, M., Kozik, R., 2020. A deep learning ensemble for network anomaly and cyber-attack detection. Sensors, 20, 4583.

[27] ElGamal, T., 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31, 469–472. doi: 10.1109/TIT.1985.1057074.

[28] Ellis, M. Christofides, P. D., 2014. Economic model predictive control with time-varying objective function for nonlinear process systems. AIChE Journal, 60(2), 507–519.

[29] Ellis, M. Christofides, P. D., 2014. Integrating dynamic economic optimization and model predictive control for optimal operation of nonlinear process systems. Control Engineering Practice, 22, 242–251.

[30] Farokhi, F., Shames, I., Batterham, N., 2017. Secure and private control using semi-homomorphic encryption. Control Engineering Practice, 67, 13–20.

[31] Gandhi, R., Sharma, A., Mahoney, W., Sousan, W., Zhu, Q., Laplante, P., 2011. Dimensions of cyber-attacks: Cultural, social, economic, and political. IEEE Technology and Society Magazine, 30, 28–38.

[32] Gentry, C., Halevi, S., Smart, N. P., 2012. Homomorphic evaluation of the aes circuit. In Annual Cryptology Conference, 850–867. Springer.

[33] Guo, Y., Hill, D. J., Wang, Y., 2000. Nonlinear decentralized control of large-scale power systems. Automatica, 36(9), 1275–1289.

[34] Hale, J. K. Lunel, S. M. V., 1993. Introduction to functional differential equations. Springer New York.

[35] Heidarinejad, M., Liu, J., Christofides, P. D., 2012. Economic model predictive control of nonlinear process systems using lyapunov techniques. AIChE Journal, 58, 855–870.

[36] Huang, L., Nguyen, X., Garofalakis, M., Hellerstein, J. M., Jordan, M. I., Joseph, A. D., Taft, N., 2007. Communication-efficient online detection of network-wide anomalies. In IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications, 134–142. IEEE.

[37] Kadakia, Y. A., Alnajdi, A., Abdullah, F., Christofides, P. D., 2023. Encrypted decentralized model predictive control of nonlinear processes with delays. Chemical Engineering Research and Design, 200, 312–324.

[38] Kadakia, Y. A., Alnajdi, A., Abdullah, F., Christofides, P. D., 2023. Encrypted distributed

model predictive control with state estimation for nonlinear processes. Digital Chemical Engineering, 9, 100133.

[39] Kadakia, Y. A., Suryavanshi, A., Alnajdi, A., Abdullah, F., Christofides, P. D., 2023. Encrypted model predictive control of a nonlinear chemical process network. Processes, 11, 2501.

[40] Kadakia, Y. A., Abdullah, F., Alnajdi, A., Christofides, P. D., 2024. Encrypted distributed model predictive control of nonlinear processes. Control Engineering Practice, 145, 105874.

[41] Kadakia, Y. A., Suryavanshi, A., Alnajdi, A., Abdullah, F., Christofides, P. D., 2024. Integrating machine learning detection and encrypted control for enhanced cybersecurity of nonlinear processes. Computers & Chemical Engineering, 180, 108498.

[42] Kagermann, H., 2014. Change through digitization—value creation in the age of industry 4.0. In Management of permanent change, 23–45. Springer.

[43] Kazantzis, N. Kravaris, C., 1998. Nonlinear observer design using lyapunov's auxiliary theorem. Systems & Control Letters, 34(5), 241–247.

[44] Khalil, H., 2002. Nonlinear Systems. Pearson Education. Prentice Hall.

[45] Khan, R., Maynard, P., McLaughlin, K., Laverty, D., Sezer, S., 2016. Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid. In Proceedings of the 4th International Symposium for ICS & SCADA Cyber Security Research, page 1–11, Belfast, United Kingdom.

[46] Kim, J., Lee, C., Shim, H., Cheon, J. H., Kim, A., Kim, M., Song, Y., 2016. Encrypting controller using fully homomorphic encryption for security of cyber-physical systems. IFAC-PapersOnLine, 49(22), 175–180.

[47] Kushner, D., 2013. The real story of stuxnet. IEEE Spectrum, 50, 48–53.

[48] Lee, J. H. Ricker, N. L., 1994. Extended kalman filter based nonlinear model predictive control. Industrial & Engineering Chemistry Research, 33(6), 1530–1541.

[49] Lin, Y. Sontag, E. D., 1991. A universal formula for stabilization with bounded controls. Systems & control letters, 16, 393–397.

[50] Liu, J., Peña, de la D. M., Ohran, B. J., Christofides, P. D., Davis, J. F., 2008. A two-tier architecture for networked process control. Chemical Engineering Science, 63(22), 5394–5409.

[51] Liu, J., Peña, Muñoz de la D., Christofides, P. D., 2009. Distributed model predictive control of nonlinear process systems. AIChE journal, 55(5), 1171–1184.

[52] Liu, J., Chen, X., Peña, Muñoz de la D., Christofides, P. D., 2010. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. AIChE Journal, 56(8), 2137–2149.

[53] Liu, J., Peña, de la D. M., Christofides, P. D., 2010. Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements. Automatica, 46(1), 52–61.

[54] Liu, J., Pena, Munoz de la D., Ohran, B. J., Christofides, P. D., Davis, J. F., 2010. A two-tier control architecture for nonlinear process systems with continuous/asynchronous feedback. International Journal of Control, 83(2), 257–272.

[55] Liu, J., Chen, X., Pena, de la D. M. M., Christofides, P. D., 2011. Iterative distributed model predictive control of nonlinear systems: Handling asynchronous, delayed measurements. IEEE Transactions on Automatic Control, 57(2), 528–534.

[56] Liu, S., Liu, J., Feng, Y., Rong, G., 2014. Performance assessment of decentralized control systems: An iterative approach. Control Engineering Practice, 22, 252–263.

[57] Luo, J. Machine Learning Modeling for Process Control and Electrochemical Reactor Operation. PhD thesis, University of California, Los Angeles, 2023.

[58] Mercorelli, P., 2017. A fault detection and data reconciliation algorithm in technical processes with the help of haar wavelets packets. Algorithms, 10(1), 13.

[59] Mhaskar, P., El-Farra, N. H., Christofides, P. D., 2006. Stabilization of nonlinear systems with state and control constraints using lyapunov-based predictive control. Systems & Control Letters, 55, 650–659.

[60] Mousavinejad, E., Yang, F., Han, Q. L., Vlacic, L., 2018. A novel cyber attack detection method in networked control systems. IEEE transactions on cybernetics, 48(11), 3254–3264.

[61] Narasimhan, S., El-Farra, N. H., Ellis, M. J., 2022. Active multiplicative cyberattack detection utilizing controller switching for process systems. Journal of Process Control, 116, 64–79.

[62] Narasimhan, S., El-Farra, N. H., Ellis, M. J., 2022. A control-switching approach for cyberattack detection in process systems with minimal false alarms. AIChE Journal, 68(12), e17875.

[63] Narasimhan, S., El-Farra, N. H., Ellis, M. J., 2023. A reachable set-based scheme for the detection of false data injection cyberattacks on dynamic processes. Digital Chemical Engineering, 7, 100100.

[64] Nedeljkovic, D. Jakovljevic, Z., 2022. Cnn based method for the development of cyberattacks detection algorithms in industrial control systems. Computers & Security, 114, 102585.

[65] Nieman, K., Messina, D., Wegener, M., Durand, H., 2023. Cybersecurity and dynamic operation in practice: Equipment impacts and safety guarantees. Journal of Loss Prevention in the Process Industries, 81, 104898.

[66] Omar, S., Ngadi, A., Jebur, H. H., 2013. Machine learning techniques for anomaly detection: an overview. International Journal of Computer Applications, 79, 33–41.

[67] Paillier, P., 1999. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the International conference on the theory and applications of cryptographic techniques, 223–238, Berlin, Heidelberg. Springer.

[68] Paridari, K., O'Mahony, N., Mady, A. E. D., Chabukswar, R., Boubekeur, M., Sandberg, H., 2017. A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration. Proceedings of the IEEE, 106(1), 113–128.

[69] Radke, A. Gao, Z., 2006. A survey of state and disturbance observers for practitioners. In 2006 American Control Conference, 5183–5188. IEEE.

[70] Rijmen, V. Daemen, J., 2001. Advanced encryption standard. Proceedings of federal information processing standards publications, national institute of standards and technology, 19, 22.

[71] Rocha, R. R., Oliveira-Lopes, L. C., Christofides, P. D., 2018. Partitioning for distributed model predictive control of nonlinear processes. Chemical Engineering Research and Design, 139, 116–135.

[72] Rossi, M. Scali, C., 2005. A comparison of techniques for automatic detection of stiction: simulation and application to industrial data. Journal of Process Control, 15(5), 505–514.

[73] Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., Harnisch, M., 2015. Industry 4.0: The future of productivity and growth in manufacturing industries. Boston consulting group, 9(1), 54–89.

[74] Scattolini, R., 2009. Architectures for distributed and hierarchical model predictive control–a review. Journal of Process Control, 19, 723–731.

[75] Schimmack, M. Mercorelli, P., 2019. An adaptive derivative estimator for fault-detection using a dynamic system with a suboptimal parameter. Algorithms, 12(5), 101.

[76] Siljak, D. D., 2011. Decentralized control of complex systems. Courier Corporation.

[77] Singh, G. G., Sajid, Z., Khan, F., Mather, C., Bernhardt, J., Frölicher, T., 2023. Rethinking disaster risk for ecological risk assessment. Frontiers in Ecology and Evolution, 11, 1249567.

[78] Smith, D. C. Cybersecurity in the energy sector: are we really prepared?, 2021.

[79] Smith, O. J., 1957. Closer control of loops with dead time. Chemical Engineering Progress, 53, 217–219.

[80] Stewart, B. T., Venkat, A. N., Rawlings, J. B., Wright, S. J., Pannocchia, G., 2010. Cooperative distributed model predictive control. Systems & Control Letters, 59(8), 460–469.

[81] Suryavanshi, A., Alnajdi, A., Alhajeri, M., Abdullah, F., Christofides, P. D., 2023. Encrypted model predictive control design for security to cyberattacks. AIChE Journal, 69, e18104.

[82] Tsvetanov, T. Slaria, S., 2021. The effect of the colonial pipeline shutdown on gasoline prices. Economics Letters, 209, 110122.

[83] Wächter, A. Biegler, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming, 106, 25–57.

[84] Wu, Z., Albalawi, F., Zhang, J., Zhang, Z., Durand, H., Christofides, P. D., 2018. Detecting and handling cyber-attacks in model predictive control of chemical processes. Mathematics, 6(10), 173.

[85] Wu, Z., Tran, A., Rincon, D., Christofides, P. D., 2019. Machine learning-based predictive control of nonlinear processes. part I: Theory. AIChE Journal, 65, e16729.

[86] Wu, Z., Tran, A., Rincon, D., Christofides, P. D., 2019. Machine learning-based predictive control of nonlinear processes. part II: Computational implementation. AIChE Journal, 65, e16734.

[87] Wu, Z., Chen, S., Rincon, D., Christofides, P. D., 2020. Post cyber-attack state reconstruction for nonlinear processes using machine learning. Chemical Engineering Research and Design, 159, 248–261.

[88] Xu, Y., Sun, Y., Wan, J., Liu, X., Song, Z., 2017. Industrial big data for fault diagnosis: Taxonomy, review, and applications. IEEE Access, 5, 17368–17380.

[89] Zeitz, M., 1987. The extended luenberger observer for nonlinear systems. Systems & Control Letters, 9, 149–156.

[90] Zhang, X. M., Han, Q. L., Ge, X., Ding, D., Ding, L., Yue, D., Peng, C., 2019. Networked control systems: A survey of trends and techniques. IEEE/CAA Journal of Automatica Sinica, 7(1), 1–17.