Computers and Chemical Engineering 200 (2025) 109166

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Encrypted machine learning-based model predictive control architectures for nonlinear systems

Arthur Khodaverdian [a], Guoquan Wu [b], Zhe Wu [b], Panagiotis D. Christofides [a,c,*]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095, USA
[b] Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore
[c] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095, USA

ARTICLE INFO

Keywords:
Encrypted control
Model predictive control
Machine learning
Nonlinear systems
Semi-homomorphic encryption

ABSTRACT

This work proposes the implementation of encryption in model predictive control of nonlinear systems in which the system dynamics are modeled through machine-learning, denoted ML-based MPC, as a means to improve cybersecurity without significant performance losses. The Pallier cryptosystem is utilized for encryption and the closed-loop stability of the encrypted ML-based MPC is established accounting for the impacts of signal quantization loss due to encryption and sample-and-hold control. A nonlinear chemical process example is used to study the impact of different encryption levels on ML-based MPC closed-loop performance. Finally, we present the implementation of the encrypted ML-based MPC method in a two-layer economic model predictive control framework and in a distributed model predictive control scheme to optimize economic performance and control large-scale processes, respectively.

1. Introduction

Model predictive control (MPC) has emerged as a powerful method for controlling nonlinear dynamic systems, which optimizes process performance while accounting for stability, actuator, and safety constraints. Traditional MPC relies on first-principles models of the system to predict future states and determine control actions. However, developing accurate first-principles models for nonlinear systems is often challenging due to the inherent complexity of system dynamics, external disturbances, and uncertainties. In recent years, MPC using machine learning (ML) to estimate the system dynamics formulation, referred to as ML-based MPC in this work, has gained significant attention as a promising alternative to address these challenges (Limon et al., 2017; Wong et al., 2018; Wu et al., 2025; Terzi et al., 2021; Hassanpour et al., 2020). Using data-driven techniques, ML models can approximate the dynamics of the system without requiring explicit knowledge of the underlying physical laws.

Although ML-based MPC demonstrates significant advantages in controlling nonlinear systems, its growing reliance on data and communication channels introduces new vulnerabilities to the stable operation of networked control systems. In modern industrial control environments, sensors, controllers, and actuators are often connected through communication networks. This enables remote access, which significantly increases the potential for cyber-attacks. Malicious cyber-attacks can compromise the integrity of sensor data, manipulate control signals, or disrupt communication channels between system components, leading to catastrophic failures in critical infrastructure (Parker et al., 2023). Specifically, attackers can exploit these vulnerabilities by injecting false data, launching denial-of-service attacks, or tampering with control actions, thereby destabilizing the closed-loop system and/or causing unsafe operations. Therefore, the development of robust cyber-secure architectures for control systems is essential to ensure stable and safe industrial operation.

In response to the increasing prevalence and threat of cyber-attacks, cybersecurity has emerged as a critical area of research in control systems (Arauz et al., 2022). Of the proposed improvements, the incorporation of encryption arguably demands the most attention, as post-quantum cryptography threatens the security of classical cryptography (Benny et al., 2024). Among the various organizations pushing for enhanced cybersecurity protocols in industry is the National Institute of Standards and Technology, which emphasizes the enhancement of data security by means of protecting confidentiality of data in use, transit, or rest (National Institute of Standards and Technology, 2024). Encrypted control systems integrate cryptographic techniques into the control process to protect sensitive data from interception or manipulation by cyber-attacks. By employing encryption schemes (e.g., homomorphic encryption), these systems allow for secure transmission of data and control actions over potentially vulnerable networks. Although homomorphic encryption allows specific operations

* Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095, USA.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.compchemeng.2025.109166
Received 31 January 2025; Received in revised form 3 April 2025; Accepted 29 April 2025
Available online 13 May 2025

(e.g., addition or multiplication) to be executed directly on encrypted data, ensuring that control actions can be executed securely without revealing the underlying data, it is insufficient for the complex, nonlinear optimization computations required by MPC. In order to encrypt in such a way that works for MPC, one could use Fully-Homomorphic encryption (FHE); however, homomorphic encryption is limited to polynomial operations, and thus all computations would need to be linearized. In addition, many FHE designs suffer from limited computational depth, significant computational overhead, compounding estimation, or quantization losses (Sui et al., 2024). Industrial applications would require real-time control which, even for linearized models, is only feasible after applying novel optimizations to the encryption scheme (Stobbe et al., 2022). Therefore, recent research has focused on developing methods to create secure connections between sensors, actuators, and control system centers, where decryption and more complex control tasks can be performed safely (Darup, 2020). The goal is to ensure that the communication between these components remains encrypted and secure throughout the entire process, from the acquisition of sensor data to the execution of control commands by the actuators.

Motivated by the above considerations, in this work, we aim to enhance the cyber-security of ML-based MPC by incorporating first-generation encrypted control techniques (Schlüter et al., 2023) in its real-time implementation. Specifically, we propose the integration of the Paillier cryptosystem into the ML-MPC framework to protect control signals and process measurement feedback from potential cyber-attacks. By analyzing the impact of quantization errors on closed-loop system, we establish the stability of the proposed closed-loop system under encrypted ML-MPC. A nonlinear chemical process is utilized to demonstrate how encryption can safeguard communication links with minimal effect on closed-loop performance. Finally, we demonstrate that the proposed encrypted ML-based MPC scheme is adaptable to various MPC frameworks, including encrypted two-layer economic MPC and distributed MPC.

2. Preliminaries

2.1. Class of systems

A class of continuous-time nonlinear systems is considered in this manuscript, which can be represented by the following ordinary differential equations:

$$\dot{x} = F(x,u) = f(x) + g(x)u \tag{1}$$

where $x \in \mathbb{R}^n$ represents the state vector and $u \in \mathbb{R}^m$ is the manipulated input vector. The manipulated input is subject to the input constraints denoted by $u \in U := \{u_i^{min} \le u_i \le u_i^{max}, \forall i = 1,2,\dots,m\} \subset \mathbb{R}^m$, where u_i^{min} and u_i^{max} are the minimum and maximum values for the manipulated input u_i. $f(\cdot)$ and $g(\cdot)$ are assumed to be sufficiently smooth functions. Additionally, it is assumed that $f(0) = 0$ such that the origin is a steady-state of the nominal system of Eq. (1).

It is assumed that there is an explicit feedback controller $u(t) = \Phi(x(t)) \in U$ that can ensure exponential stability of the origin of Eq. (1). The stabilizability assumption implies the existence of a continuously differentiable control Lyapunov function $V(x)$ such that the following inequalities hold for all $x \in D$, where D is an open neighborhood around the origin:

$$c_1|x|^2 \le V(x) \le c_2|x|^2 \tag{2a}$$

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

$$\left|\frac{\partial V(x)}{\partial x}\right| \le c_4|x| \tag{2c}$$

where c_1, c_2, c_3, and c_4 denote positive constants. Furthermore, due to the Lipschitz continuity of $F(x,u)$ and the fact that u is bounded, there exist positive constants M_F, L_x, and L_x' that ensure that for all $x, x' \in D$ and $u \in U$ the following inequalities are satisfied:

$$|F(x',u) - F(x,u)| \le L_x|x - x'| \tag{3a}$$

$$|F(x,u)| \le M_F \tag{3b}$$

$$\left|\frac{\partial V(x)}{\partial x} F(x,u) - \frac{\partial V(x')}{\partial x} F(x',u)\right| \le L_x'|x - x'| \tag{3c}$$

2.2. Paillier cryptosystem and quantization

The Paillier cryptosystem (Paillier, 1999) is an additive homomorphic encryption algorithm that allows an arbitrary number of addition operations to be performed in the encrypted space. The public and private keys used to encrypt and decrypt the ciphertext, respectively, can be calculated through the following set of steps:

1. Generate two prime integers denoted p and q randomly.
2. Verify that $\gcd(pq, (p-1)(q-1)) = 1$ before proceeding, where $\gcd(i,j)$ is a function that returns the greatest common divisor of natural numbers i, j. Repeat 1 as needed.
3. Define $M = pq$ to simplify the notation.
4. Randomly generate an integer $g \in \mathbb{Z}_{M^2}^*$.
5. Define $\lambda = \mathrm{lcm}(q-1, p-1)$, where $\mathrm{lcm}(i,j)$ refers to the least common multiple of the integers i, j.
6. Define a function $\bar{L}(x) = (x-1)/M$ to simplify the notation.
7. Ensure that the following modular multiplicative inverse exists before proceeding:
   $u = (\bar{L}(g^\lambda \bmod M^2))^{-1} \bmod M$. Repeat 4 as needed.
8. (M, g) and (λ, u) are the public and private keys, respectively.

Encryption is performed using Eq. (4).

$$E_M(m,r) = c = g^m r^M \bmod M^2 \tag{4}$$

where $r \in \mathbb{Z}_M$ and is a randomly generated scalar. $\mathbb{Z}_{M^2}^*$ denotes the multiplicative group of integers modulo M^2. c is called ciphertext, and is the encrypted form of plaintext m. Decryption is done using Eq. (5).

$$D_M(c) = m = \bar{L}(c^\lambda \bmod M^2) u \bmod M \tag{5}$$

Paillier encryption only works for non-negative integers, but the cryptosystem can be modified to work for arbitrary scalars by quantizing and mapping (Darup et al., 2017). Quantization maps from the set of real-numbers $\mathbb{R}$ to the set of signed fixed-point binary numbers, $\mathbb{Q}_{l,d}$ where l is the total bit length l and d is the fractional bit length. This set can be represented using the two's complement (Intel, 2024) representation as $\mathbb{Q}_{l,d} = \{q \in \mathbb{Q} | q = -2^{l-d-1}\beta_l + \sum_{i=1}^{l-1} 2^{i-d-1}\beta_i, \beta_i \in \{0,1\} \forall i = 1 \cdots l\}$. Quantization can be done with Eq. (6):

$$\begin{aligned} &g_{l,d} : \mathbb{R} \to \mathbb{Q}_{l,d} \\ &g_{l,d}(a) := \arg\min_{q \in \mathbb{Q}_{l,d}} |a - q| \end{aligned} \tag{6}$$

The quantized data can now be bijectively mapped to a subset of non-negative integers (Darup et al., 2017). This subset is called the message space $\mathbb{Z}_M$:

$$\begin{aligned} &f_{M,d} : \mathbb{Q}_{l,d} \to \mathbb{Z}_M \\ &f_{M,d}(q) := 2^d q \bmod M \end{aligned} \tag{7}$$

To recover data after decryption is done, the mapping must be undone as follows:

$$f_{M,d}^{-1} : \mathbb{Z}_M \to \mathbb{Q}_{l,d} \tag{8}$$

$$f_{M,d}^{-1}(m) := \frac{1}{2^d} \begin{cases} m - M & \text{if } m \ge M - M//3 + 1 \\ m & \text{otherwise} \end{cases} \tag{9}$$

In Eq. (9), the '//' operation is known as integer division. Unlike traditional division, integer division truncates the decimal values. In

Python, negative results are instead floored, or truncated away from 0. In this particular encoding of Pallier Encryption, roughly 1/3 of the message space is allocated solely to detecting overflow. Because the positive quadrant of the message space includes 0, the effective maximum integer stored in the message space is $M//3-1$, and thus the negative quadrant begins at $M-(M//3-1)$ as shown above.

Overflow occurs when values become so large in the message space that they move from the positive subset to the negative subset, which is possible due to the bijective mapping of negative values to the upper half of the message space. This can occur from arithmetic operations that gradually increase the precision through subsequent multiplication operations or from operations that yield large numbers that cannot be contained within the defined message space. Thus, by allocating a third of the message space as invalid, there exists a buffer for detecting overflow at the expense of shrinking the usable message space.

Addition can be done using Eq. (10).

$$D_M((E_M(m_1,r_1)\cdot E_M(m_2,r_2))\mathrm{mod}M^2)=(m_1+m_2)\mathrm{mod}M \tag{10}$$

Note that the bijective mapping from earlier can map the same number to different values in the message space if the fractional bit length d is different. This can cause issues when adding numbers, so the two terms are remapped to the larger of their respective d values prior to adding. To allow for this, d is tracked for encrypted objects.

$$m_{d_2}=m_{d_1}2^{d_1-d_2}:d_1<d_2 \tag{11}$$

Multiplication can be done with Eq. (12).

$$D_M(E_M(m_1,r_1)^k\mathrm{mod}M^2)=km_1\mathrm{mod}M \tag{12}$$

Unlike addition, the exponent is plaintext, meaning one of the terms must be plaintext for multiplication to occur. If we substitute Eq. (7) into the plaintext forms it can be seen that multiplication yields a new d equivalent to adding the respective d values from the two multiplied terms.

3. Encrypted ML-based MPC

3.1. ML modeling of nonlinear systems

Given the high degree of coupling and complexity of nonlinear systems, deriving first-principles models for such systems may be impractical or infeasible in many real-world applications. To address this limitation, ML techniques, particularly recurrent neural networks (RNNs), have emerged as powerful tools for capturing the behavior of nonlinear systems using time-series data. Specifically, RNNs are well suited for modeling dynamic systems due to their ability to capture temporal dependencies and process sequential data. Unlike feedforward neural networks, which assume independence between inputs, RNNs use internal memory to store information about previous inputs, allowing them to learn from past states and predict future system behavior. The formulation of RNNs can be presented as follows:

$$h_t=\sigma_h\left(W_h h_{t-1}+W_i z_t\right) \tag{13a}$$

$$\hat{y}_t=\sigma_y\left(W_o h_t\right) \tag{13b}$$

where z_t represents the RNN input, h_t represents the hidden states, $\hat{y}_t$ represents the output, The weight matrices W_i, W_o, and W_h are associated with the input vector, output layer, and hidden states, respectively. The activation functions applied to each element in the hidden and output layers are represented by σ_h and σ_y, respectively.

In this work, the RNN is constructed to capture the process dynamic behavior of the system of Eq. (1), serving as the predictive model for the MPC framework. Specifically, the RNN receives the current state $x(t)$ and manipulated input $u(t)$ at $t=t_k$ to predict the future state of the nonlinear system of Eq. (1) over $t\in\left[t_k,t_k+N\Delta\right]$, where Δ is the sampling period and N is the prediction horizon. To accomplish this, a dataset of state trajectories under randomized initial conditions and manipulated inputs within their operating constraints is first collected. Next, the dataset is preprocessed to ensure consistency and improve the learning process by using standard scalers for each dimension. Additionally, the dataset is segmented into training, validation, and testing sets to evaluate the performance of the model and avoid overfitting. Finally, a gradient-based optimizer (e.g., Adam) is employed to minimize the loss function (i.e., mean squared error (MSE)), which quantifies the difference between the predicted and true system states.

It is assumed that there is a stabilizing feedback controller $u=\boldsymbol{\Phi}_{nn}(x)\in U$ for the RNN model, which is a Lipschitz continuous function and can ensure exponential stability of the origin within an open neighborhood $\hat{D}$ around the origin. This implies that there exists a C^1 Control Lyapunov function $V(x)$ such that the following inequalities are satisfied for all x in $\hat{D}$:

$$\hat{c}_1|x|^2\le V(x)\le\hat{c}_2|x|^2 \tag{14a}$$

$$\frac{\partial V(x)}{\partial x}F_{nn}\left(x,\boldsymbol{\Phi}_{nn}(x)\right)\le-\hat{c}_3|x|^2 \tag{14b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right|\le\hat{c}_4|x| \tag{14c}$$

where $\hat{c}_1,\hat{c}_2,\hat{c}_3,\hat{c}_4$ are positive constants, and F_{nn} is the RNN model. The stability region Ω_ρ is characterized as a level set of V within the set where $\dot{V}\le-kV(x)$ is satisfied using $u=\boldsymbol{\Phi}_{nn}(x)\in U$. While the Lyapunov function designed for the RNN model may not be the same as that for Eq. (1), we will use the same notation, V, throughout this manuscript for simplicity, with a slight abuse of notation.

Remark 1. In this work, we consider traditional machine learning methods that train an RNN model on a central server using historical process operation data. However, it should be noted that there may be potential data security and privacy issues during the modeling stage, particularly for modeling nonlinear distributed systems. Traditional centralized training methods require the aggregation of all data on a central server for processing, which can expose data to insecure networks during transmission or storage, increasing the risk of data breaches or unauthorized access. One promising approach to address this issue is federated learning (Xu and Wu, 2024), a decentralized learning technique that allows individual systems to collaboratively train ML models without sharing raw data. Federated learning keeps data locally, exchanging only model weights between subsystems, and therefore ensures data privacy in modeling of large-scale systems.

3.2. ML-based MPC

In this work, we develop an encrypted ML-based MPC framework for the nonlinear system of Eq. (1), which incorporates secure communication between sensors, controllers, and actuators. The goal is to optimize closed-loop performance under MPC while ensuring data security across communication links. As shown in Fig. 1, the sensor measurements $x(t)$ are encrypted and transmitted to the MPC to prevent eavesdropping or tampering. The encrypted states are then decrypted to obtain the quantized states $\hat{x}(t)$, which are then used by the ML model within the MPC to predict future states over a prediction horizon Δ. The MPC solves an optimization problem at each time step to compute the optimal manipulated inputs $u(t)$, which are then encrypted before transmitting to the actuator. At the actuator, the encrypted manipulated input signals are decrypted to obtain the quantized control signal $\hat{u}(t)$, which is implemented to the nonlinear process. The proposed closed-loop system secures both sensor-to-controller and controller-to-actuator communication while maintaining the desired control performance.

Specifically, the optimization problem of the ML-based MPC is shown as follows:

$$\mathcal{J}=\min_{u\in S(\Delta)}\int_{t_k}^{t_{k+N}}L(\tilde{x}(t),u(t))dt \tag{15a}$$

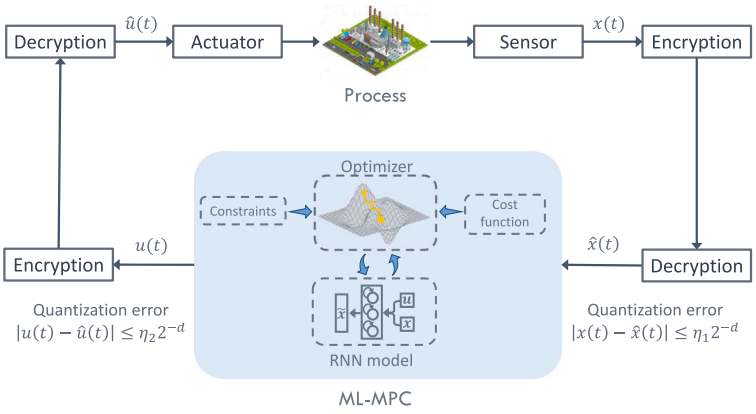

Fig. 1. Framework of closed-loop system under encrypted ML-based MPC.

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), u(t)) \tag{15b}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{15c}$$

$$u(t) \in U, \quad \forall t \in [t_k, t_{k+N}) \tag{15d}$$

$$\dot{V}\left(\hat{x}(t_k), u(t_k)\right) \leq \dot{V}\left(\hat{x}(t_k), \Phi_{nn}\left(\hat{x}(t_k)\right)\right), \quad \text{if } \hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{15e}$$

$$V(\tilde{x}(t)) \leq \rho_{\min}, \quad \forall t \in [t_k, t_{k+N}), \quad \text{if } \hat{x}(t_k) \in \Omega_{\rho_{\min}} \tag{15f}$$

where $\tilde{x}$ represent the predicted state trajectory, $S(\Delta)$ represent the set of piecewise constant functions with the sampling period Δ, N is the number of sampling periods in the prediction horizon, and $\dot{V}(x,u)$ is the time derivative of $V(x)$, i.e., $\frac{\partial V}{\partial x} F_{nn}(x,u)$. The objective function of Eq. (15a) is formulated by minimizing the integral of the cost function $L(\tilde{x}(t), u(t))$ over the prediction horizon. The RNN model F_{nn} is the predictive model for MPC to optimize control actions. The constraints specified in Eqs. (15e) and (15f) are imposed to guarantee the closed-loop stability within the defined stability region.

3.3. Closed-loop stability of encrypted ML-MPC

In the closed-loop design illustrated in Fig. 1, there are sources of error. Specifically, state quantization errors exist in the sensor-controller communication link, and input quantization errors occur in the controller-actuator communication link. According to the mapping in Eq. (6), these quantization errors are bounded as follows:

$$|x(t) - \hat{x}(t)| \leq \eta_1 2^{-d} \tag{16a}$$

$$|u(t) - \hat{u}(t)| \leq \eta_2 2^{-d} \tag{16b}$$

where $0 \leq \eta_1, \eta_2 \leq 0.5$, and d represent the quantization parameter in the mapping described by Eq. (6). Quantization rounds numbers to the nearest element in the set $\mathbb{Q}_{l,d}$, which is equivalent to rounding to the nearest number that is divisible by the precision term 2^{-d}. Thus, the quantization error is bounded by the worst-case rounding error, which is the magnitude of this precision term. The coefficients η_1 and η_2 account for the non worst-case errors. Given the quantization error introduced into the input, the nonlinear system of Eq. (1) in the closed-loop design of Fig. 1 can be presented as follows:

$$\begin{aligned} \dot{x} &= F(x, \hat{u}) = f(x) + g(x)\hat{u} \\ &= f(x) + g(x)\left(u + e_2\right) \end{aligned} \tag{17}$$

where $e_2 = \hat{u}(t) - u(t)$ and $|e_2| \leq \eta_2 2^{-d}$. Additionally, an error occurs in the computed manipulated inputs because the MPC processes the quantized state $\hat{x}$ instead of the actual state x. Based on the stabilizing control law $u = \Phi_{nn}(x) \in U$, the resulting error in the control input remains bounded as follows:

$$|\Phi_{nn}(\hat{x}) - \Phi_{nn}(x)| \leq L_1 |\hat{x} - x| \leq L_1' 2^{-d} \tag{18}$$

where L_1 is the Lipschitz constant for the controller $\Phi_{nn}(x)$ and $L_1' = L_1 \eta_1$. Additionally, to demonstrate that the origin of Eq. (1) can be rendered exponentially stable $\forall x \in \Omega_\rho$ under the controller $\Phi_{nn}(x)$ designed for the RNN model, we prove that $\dot{V}$ for the system of Eq. (1) can still be rendered negative under $u = \Phi_{nn}(x) \in U$. Based on Eq. (14b), the time-derivative of V is derived as follows:

$$\begin{aligned} \dot{V} &= \frac{\partial V(x)}{\partial x} F\left(x, \Phi_{nn}(x)\right) \\ &= \frac{\partial V(x)}{\partial x}\left(F_{nn}\left(x, \Phi_{nn}(x)\right) + F\left(x, \Phi_{nn}(x)\right) - F_{nn}\left(x, \Phi_{nn}(x)\right)\right) \\ &\leq -\hat{c}_3 |x|^2 + \hat{c}_4 |x| \left(F\left(x, \Phi_{nn}(x)\right) - F_{nn}\left(x, \Phi_{nn}(x)\right)\right) \end{aligned} \tag{19}$$

If the modeling error is trained to satisfy $|F\left(x, \Phi_{nn}(x)\right) - F_{nn}(x, \Phi_{nn}(x))| \leq \beta |x|$ for all states and inputs, and β satisfies $\beta < \hat{c}_3 / \hat{c}_4$, then we have $\dot{V} \leq -\tilde{c}_3 |x|^2 \leq 0$ where $\tilde{c}_3 = -\hat{c}_3 + \hat{c}_4 \beta > 0$. Therefore, the closed-loop state of the nonlinear system of Eq. (1) converges to the origin under the control law $u = \Phi_{nn}(x) \in U$ for all $x_0 \in \Omega_\rho$.

Theorem 1. *Consider the nonlinear system of Eq. (1) with an initial state $x_0 \in \Omega_\rho$. The origin of Eq. (17) is practically stable under encrypted ML-MPC designed with the stabilizing control law $u = \Phi_{nn}(x) \in U$. This implies that for all $x_0 \in \Omega_\rho$, the closed-loop state $x(t)$ remains within Ω_ρ at all times, and the following inequalities are satisfied:*

$$\dot{V} \leq -c_5 |x|^2, \quad \forall\ |x| \geq \frac{c_4 2^{-d}\left(\gamma_1 + \gamma_2\right)}{\tilde{c}_3 \theta} = \mu \tag{20a}$$

$$\limsup_{t \to \infty} |x(t)| \leq b \tag{20b}$$

Proof. The time derivative of V can be expressed as follows based on the nonlinear system of Eq. (17):

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial x} F(x, \hat{u}) \\ &= \frac{\partial V}{\partial x} F\left(x, \Phi_{nn}(\hat{x}) + e_2\right) \\ &= \frac{\partial V}{\partial x}\left[f(x) + g(x)\left(\Phi_{nn}(\hat{x}) + e_2\right)\right] \\ &= \frac{\partial V}{\partial x}\left[f(x) + g(x)\left(\Phi_{nn}(\hat{x}) - \Phi_{nn}(x) + \Phi_{nn}(x) + e_2\right)\right] \\ &= \frac{\partial V}{\partial x}(f(x) + g(x)\Phi_{nn}(x)) + \frac{\partial V}{\partial x} g(x)(\Phi_{nn}(\hat{x}) - \Phi_{nn}(x)) + \frac{\partial V}{\partial x} g(x) e_2 \end{aligned} \tag{21}$$

The following expression can be derived based on Eq. (19):

$$\dot{V} \leq -\tilde{c}_3 |x|^2 + \frac{\partial V}{\partial x} g(x)(\Phi_{nn}(\hat{x}) - \Phi_{nn}(x)) + \frac{\partial V}{\partial x} g(x) e_2 \tag{22}$$

By applying the inequalities from Eqs. (2b), (16), (17), and (19), the following result can be obtained:

$$\begin{aligned}\dot{V} &\leq -\tilde{c}_3|x|^2 + c_4\gamma_1|x|2^{-d} + c_4\gamma_2|x|2^{-d} \\ &\leq -\tilde{c}_3|x|^2 + c_4|x|2^{-d}\left(\gamma_1+\gamma_2\right) \\ &\leq -(1-\theta)\tilde{c}_3|x|^2 - \theta\tilde{c}_3|x|^2 + c_4 2^{-d}\left(\gamma_1+\gamma_2\right)|x|\end{aligned} \tag{23}$$

where $g(x)$ in Eq. (22) is assumed to be bounded by γ, $\forall x \in \Omega_\rho$, $\gamma > 0$, $\gamma_1 = \gamma L_1'$, and $\gamma_2 = \gamma\eta_2$. Therefore, if the condition of Eq. (20a) on $|x|$ is satisfied, that is, $|x| \geq \frac{c_4 2^{-d(\gamma_1+\gamma_2)}}{\tilde{c}_3\theta} = \mu$, the following holds:

$$\dot{V} \leq -(1-\theta)\tilde{c}_3|x|^2 \leq -c_5|x|^2 \tag{24}$$

where $c_5 = (1-\theta)\tilde{c}_3$. Therefore, it follows that $\dot{V}$ is negative for all $x \in \Omega_\rho$ that meet the condition specified in Eq. (20a) based on Eq. (24).

Given that Ω_ρ is a level set of V and that $\dot{V}$ is negative for all $x \in \Omega_\rho$, it follows that the closed-loop state $x(t)$ remains within Ω_ρ at all times. Additionally, by applying theorem 4.18 from Ref. Khalil (2002), it can be derived that:

$$\limsup_{t\to\infty} |x(t)| \leq b \tag{25}$$

where b is a positive constant that can be represented as a class $\mathcal{K}$ function of μ. As the quantization parameter d increases towards infinity, μ approaches zero based on its definition in Eq. (20a). Therefore, the ultimate bound converges to zero, indicating that larger values of d reduce the error between the state and input trajectories of the encrypted and non-encrypted control systems. This confirms that, for sufficiently large d, the closed-loop states of Eq. (17) are ultimately uniformly bounded under $u = \Phi_{nn}(x) \in U$. □

Unlike uncertainties, quantization losses due to encryption are both strictly bounded and adjustable. Theorem 1 demonstrates how stability is guaranteed when applying ML-based Lyapunov-based control assuming a sufficiently low modeling error in the dynamics RNN model and an encryption level that can be tolerated by the closed-loop system under ML-based Lyapunov-based control. In other words, through the analysis, given the robustness margin, expressed in terms of the bound in the Lyapunov function derivative along the trajectory of the closed-loop system under ML-based Lyapunov-based control, one can determine the encryption level that leads to an additional error contribution for which closed-loop stability is maintained. This result is used to establish closed-loop stability under encrypted ML-MPC in Theorem 2 below.

Theorem 2. *Consider the nonlinear system of Eq. (1), operating under the closed-loop encrypted LMPC framework with the stabilizing controller $u = \Phi_{nn}(x) \in U$ and the initial state $x_0 \in \Omega_\rho$. Let $\epsilon_w > 0$, $\Delta > 0$ and $\rho > \rho_{min} > \rho_s$ satisfy,*

$$\rho_{min} = \max\{V(x(t+\Delta)) \mid V(x(t)) \leq \rho_s\} \tag{26a}$$

$$\frac{-\tilde{c}_3}{c_2}\rho_s + L_x' M_F \Delta + L_w' \delta s \leq -\epsilon_w \tag{26b}$$

where $|e_2| \leq \eta_2 2^{-d} = \delta$ and $L_w' > 0$. Then, the closed-loop state $x(t)$ is always bounded in Ω_ρ and is ultimately bounded in $\Omega_{\rho_{min}}$.

Proof. The proof closely follows the approach used for Theorem 2 in Suryavanshi et al. (2023) and involves the following steps. Consider the state $x_{t_k} \in \Omega_\rho \backslash \Omega_{\rho_s}$. The time-derivative of V under the control inputs calculated by the LMPC of Eq. (15) for the nonlinear system of Eq. (17) at t_k can be written as:

$$\begin{aligned}\dot{V} =& \frac{\partial V(x(t))}{\partial x} F\left(x(t), u(t_k), e_2\right) \\ =& \frac{\partial V\left(x_{t_k}\right)}{\partial x} F\left(x_{t_k}, u(t_k)\right) + \frac{\partial V(x(t))}{\partial x} F\left(x(t), u(t_k), e_2\right) \\ &- \frac{\partial V\left(x_{t_k}\right)}{\partial x} F\left(x_{t_k}, u(t_k)\right)\end{aligned} \tag{27}$$

In the encrypted LMPC, the constraint of Eq. (15e) ensures that, if $x_{t_k} \in \Omega_\rho \backslash \Omega_{\rho_{\min}}$, then the closed-loop state is driven towards the origin at t_{k+1} (to a lower level set of V). Based on the inequality of Eq. (2b), it follows from Eq. (27) that:

$$\begin{aligned}\dot{V} \leq& -c_3\left|x_{t_k}\right|^2 + \frac{\partial V(x(t))}{\partial x} F\left(x(t), u(t_k), e_2\right) \\ &- \frac{\partial V\left(x_{t_k}\right)}{\partial x} F\left(x_{t_k}, u(t_k)\right)\end{aligned} \tag{28}$$

Based on the fact that the error, $|e_2| \leq \eta_2 2^{-d} = \delta$ is bounded, the Lipschitz conditions of Eq. (3), and the inequality of Eq. (2a), it follows from Eq. (28) that:

$$\dot{V} \leq -\frac{c_3}{c_2}\rho_s + L_x'|x(t) - x_{t_k}| + L_w'\delta \tag{29}$$

where $L_w' > 0$. 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. (29) that:

$$\dot{V} \leq -\frac{c_3}{c_2}\rho_s + L_x' M_F \Delta + L_w'\delta \tag{30}$$

Thus, if $\frac{-c_3}{c_2}\rho_s + L_x' M_F\Delta + L_w'\delta s \leq -\epsilon_w$, then $\dot{V} \leq -\epsilon_w$ for any $x_{t_k} \in \Omega_\rho \setminus \Omega_{\rho_s}$. This establishes that the state of the closed-loop system is always bounded in Ω_ρ, and it ultimately converges to $\Omega_{\rho_s} \subseteq \Omega_{\rho_{\min}}$ and then remains there. □

4. Application to a chemical reactor

A non-isothermal continuous stirred tank reactor (CSTR), assumed to be well-mixed, is considered for the simulation. The reactor has an irreversible, second-order, exothermic reaction in which a reactant A is transformed into product B ($A \rightarrow B$) in the CSTR. Heat is added or removed via a heating jacket at a rate Q. The dynamic model of the CSTR is expressed as follows:

$$\begin{aligned}\frac{dC_A}{dt} &= \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 \\ \frac{dT}{dt} &= \frac{F}{V}(T_0 - T) - \frac{\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V}\end{aligned} \tag{31}$$

where C_{A0} is the feed concentration of reactant A, Q is the rate of heat input, C_A is the concentration of reactant A, and T is the reactor temperature.

4.1. Control problem formulation and MPC design

The rate of heat input (Q) and the feed concentration of reactant A (C_{A0}) are chosen as the control inputs. The state variables and control inputs are defined in deviation variable form relative to their steady state. The resulting state-space is defined as $x = [C_A - C_{A_s}, T - T_s]$, and the resulting control input variables are defined as $u = [C_{A0} - C_{A0_s}, Q - Q_s]$. The control inputs are bounded such that $[-3.5\,\frac{\text{kmol}}{\text{m}^3} \leq C_{A0} \leq 3.5\,\frac{\text{kmol}}{\text{m}^3}]$ and $[-5\times10^5\,\frac{\text{kJ}}{\text{h}} \leq Q \leq 5\times10^5\,\frac{\text{kJ}}{\text{h}}]$. The value of the control inputs are dictated by the MPC defined in Eq. (15). Relevant parameters used in the dynamic model are shown in Table 1.

To demonstrate the importance of a high precision parameter, the system will be designed with 3 variants: an unencrypted control system, a system with $d = 4$ precision, and a system with $d = 12$ precision. As discussed in Section 2.2, real numbers are not natively supported, which is why data must be quantized and mapped prior to encryption. The precision coefficient d implies that our data is rounded to the nearest 2^{-d} which introduces an error upwards of 2^{-d-1} to any encrypted values.

The MPC for this system operates with a sampling time of 0.01 h (36 s) over 12 iterations for a total operating time of 0.12 h. Because no physical plant was used, the sensor readings were calculated using the dynamic model shown in Eq. (31) with the parameters shown in Table 1. Integration was performed with the forward Euler method with an integration step size of $h_c = 1\times10^{-4}$ h. The MPC uses the ML model to

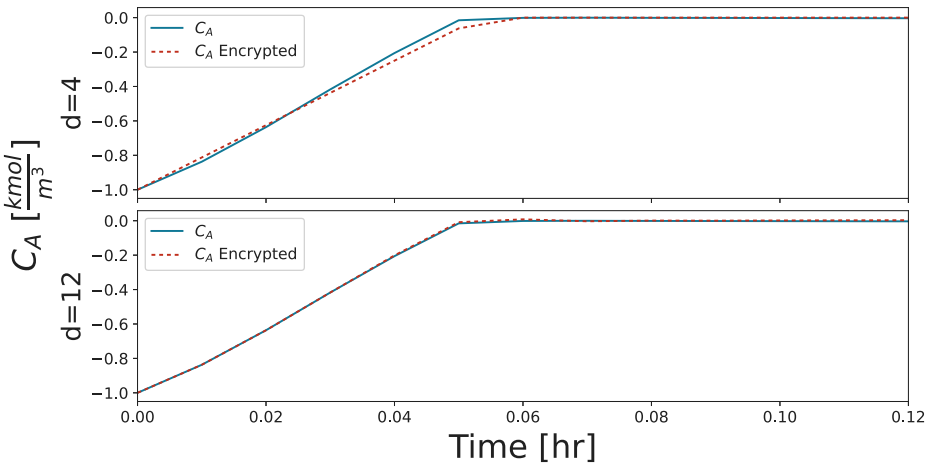

Fig. 2. Impact of (d = 4) precision compared to (d = 12) precision for concentration.

Table 1
Parameter values for the chemical process example.

Name	Label	Value	Units
Flow rate	F	5	m^3/h
Reactor volume	V	1	m^3
Pre-exponential factor	k_0	8.46e6	$m^3/(kmol\ h)$
Activation energy	E	5e4	kJ/kmol
Gas constant	R	8.314	kJ/(kmol K)
Liquid density	ρ_L	1000	kg/m^3
Enthalpy of reaction	ΔH	−1.15e4	kJ/kmol
Inlet temperature	T_0	300	K
Steady-state heat input rate	Q_s	0	kJ/h
Steady-state feed concentration	C_{A0_s}	4	$kmol/m^3$
Steady-state concentration	C_{As}	1.9537	$kmol/m^3$
Steady-state temperature	T_s	401.8727	K
Specific heat	C_p	0.231	kJ/(kg K)

calculate its state trajectory based on the current state measurements from the simulation of the dynamic model. The MPC uses a prediction horizon of 2 sampling periods. The Lyapunov function used to ensure stability is of the form $V = x^T Px$ where P is defined in Eq. (32):

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{32}$$

Using this Lyapunov function, we define the two operating modes based on if the value of this function exceeds $\rho_{min} = 2$. If true, we enforce Eq. (15e) where $\dot{V}$ is required to be less than $-kV(\tilde{x}(t))$, where $k > 0$, since the stability region is characterized for the set of states that satisfy $\dot{V} \leq -kV(\tilde{x}(t))$ for all states within Ω_ρ. If false, then we enforce Eq. (15f). The objective function for MPC is chosen to be:

$$L(\tilde{x}(t), u(t)) = (T - T_s)^2 + 1000(C_A - C_{A_s})^2 \tag{33}$$

Remark 2. IPOPT is used as the optimizer in the MPC. A tolerance of $1e-5$ was used with a max of 1000 iterations to allow for a sufficiently locally optimal solution to be found. The necessary gradients for the optimizer are solved numerically with a step-size of 0.1. The resulting MPC solutions are found within a sufficiently fast time frame such that the total computation time, including encryption and decryption is less than the sampling time of the process. Data samples are generated by running open-loop simulations with various initial states and manipulated inputs following the method in our previous work (Wu et al., 2019). Specifically, we collected approximately 10,000 data samples by evenly distributing data points for two states – reactant concentration and reactor temperature – and two manipulated inputs – inlet reactant concentration and heat input rate – within their respective ranges. The RNN model achieves a sufficiently small modeling error of approximately 10^{-5}. The model consists of two hidden layers with 128 and 64 neurons, respectively, and employs the Adam optimizer for training. The mean squared error (MSE) is used as the loss function to minimize prediction errors. To prevent overfitting, we apply an early stopping mechanism, which monitors the validation loss and halts training when no significant improvement is observed.

Remark 3. The encryption method of choice is the Pallier cryptosystem, although other encryption methods will also work for the given design as no arithmetic operations are done on the encrypted data.

Remark 4. The implementation of the Pallier cryptosystem is done by the Python Pallier package (Data61, 2013), which uses base 16 precision as opposed to our base 2 precision. This is why the small precision coefficient d is chosen as 4 instead of 1, as a value of 1 would be rounded to 4 because the base 16 exponent is calculated by $\lfloor \log_{16} 2^{-1} \rfloor = -1$ and thus the precision is $16^{-1} = 2^{-4}$.

4.2. Simulation results

As shown in Figs. 2 to 5, the impact of encryption is initially negligible, but as time goes on the system begins to deviate from the nominal closed-loop trajectories; the light blue curves in Figs. 2 to 5 correspond to the closed-loop states and the manipulated input profiles under the ML-MPC system in the absence of encryption. Although the overall behavior of each term remains similar, the quantization errors can compound with system dynamics to produce larger errors than what might be expected. This is best seen in Fig. 4 where the $d = 4$ precision plot produces an input at $t = 0.05$ h that is visibly larger than the unencrypted system, whereas the $d = 12$ precision plot yields a far more similar input to the unencrypted system. Furthermore, Figs. 2 to 3 show that the impact of quantization errors is minimal with respect to tracking control as neither plot shows oscillatory behavior, offset tracking, or delayed convergence. The primary impact of quantization errors is also seen in the control inputs as discussed earlier.

5. Two-layer architecture and distributed scheme of encrypted ML-based MPC

In this section, we demonstrate that the proposed encrypted ML-based MPC method can be applied to various control schemes with different control objectives. Specifically, we consider the two-layer control framework and the iterative distributed scheme that optimize process economic performance and control large-scale nonlinear systems, respectively.

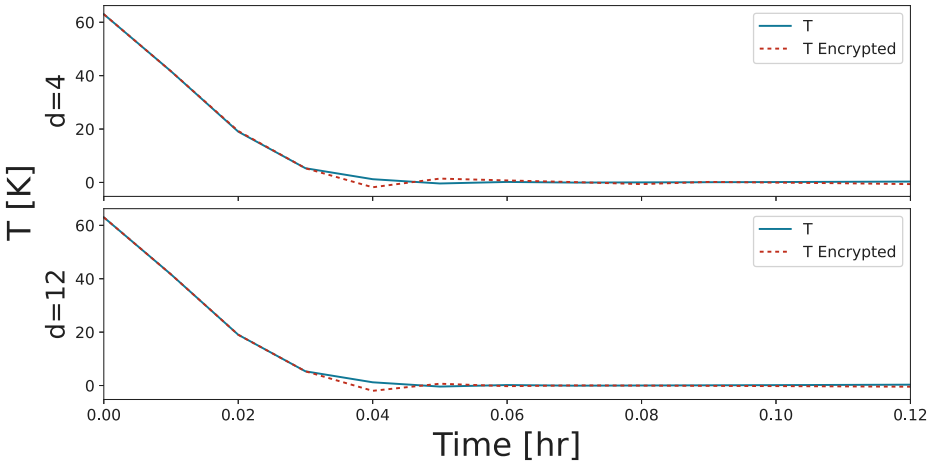

Fig. 3. Impact of (d = 4) precision compared to (d = 12) precision for temperature.

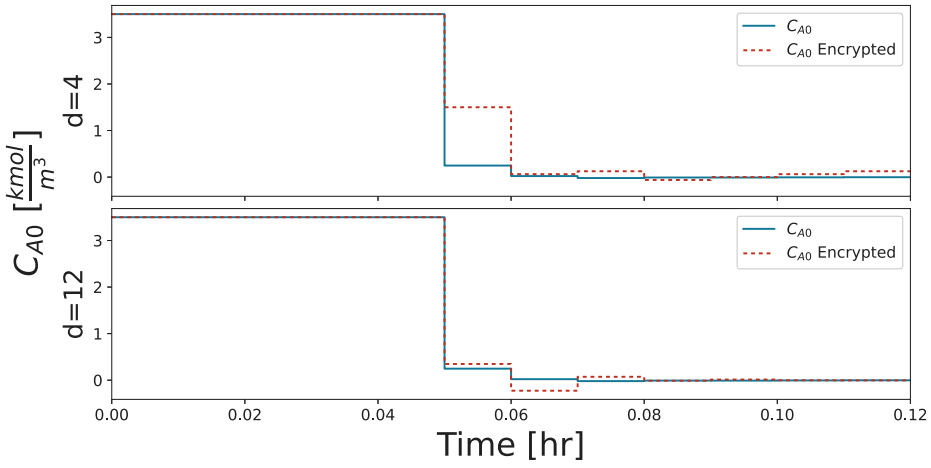

Fig. 4. Impact of (d = 4) precision compared to (d = 12) precision for inlet concentration.

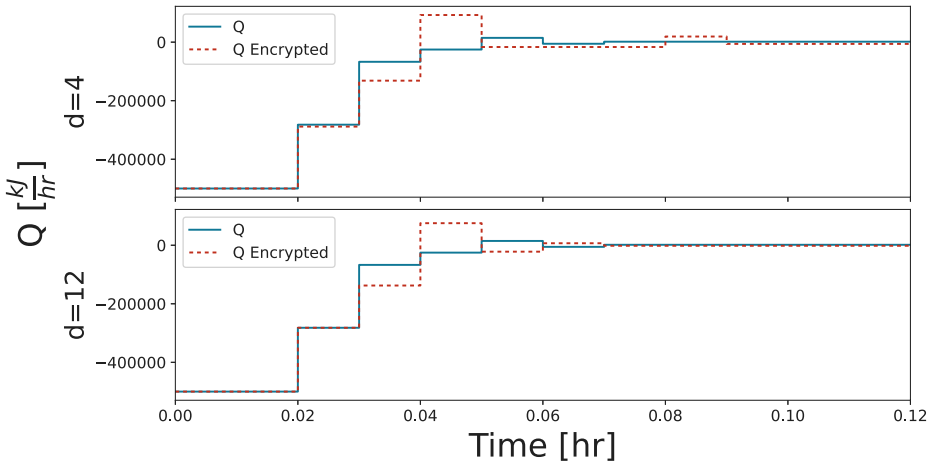

Fig. 5. Impact of (d = 4) precision compared to (d = 12) precision for heat input.

5.1. Two-layer architecture of encrypted ML-based EMPC

Traditional tracking MPCs are equipped with an upper-tier optimizer that determines the optimal steady-states to maximize the economic profits (e.g., energy consumption and efficiency). Economic MPC (EMPC) can be utilized to improve economic performance while ensuring closed-loop stability. In this subsection, the proposed encrypted ML-based MPC is incorporated into the two-layer architecture (see Fig. 6) to optimize economic performance while maintaining the security and stability of the closed-loop system.

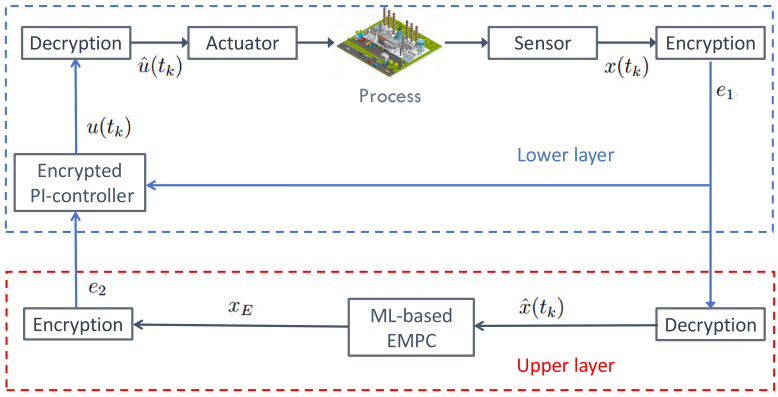

Fig. 6. Encrypted ML-based EMPC with two-layer architecture.

The encrypted ML-based EMPC scheme in Fig. 6 consists of two layers, where the upper layer contains the EMPC using the decrypted plaintext, and the lower layer is with a set of linear controllers (i.e., PI controllers) using encrypted information. The communication channels using encrypted data, as illustrated in Fig. 6, are highlighted in blue. Specifically, in Fig. 1, at time t_k, the sensor signal $x(t_k)$ is encrypted as e_1, and then the ciphertext e_1 is transmitted to the ML-based EMPC, where the state information is decrypted to plaintext $\hat{x}(t_k)$. The optimal state trajectory over $[t_k, t_k + \Delta)$ is calculated in the EMPC using the decrypted state signal $\hat{x}(t_k)$, and the machine learning model is used to predict the system dynamics. Next, the optimal state trajectory obtained by the EMPC is encrypted and transmitted to the lower layer with linear controllers. Due to the homomorphic property of the Pailier cryptosystem, the linear PI controller is selected to generate the control input signal in the encrypted space with the encrypted set point and encrypted state signals. The encrypted control input is transmitted to the actuator, which has access to the private key to decrypt the input signal to plaintext $\hat{u}(t_k)$. The two controllers in the upper and lower layers are working together to improve the economic performance of the encrypted ML-based EMPC scheme while ensuring the security and stability of the closed-loop system.

The EMPC in the upper layer can be represented by the following optimization problem:

$$\mathcal{J} = \max_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L_e(\tilde{x}(t), u(t))\, \mathrm{d}t \tag{34a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), u(t)) \tag{34b}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{34c}$$

$$u \in U, \ \forall\, t \in [t_k, t_{k+N}) \tag{34d}$$

$$|\dot{\tilde{x}}(t)| \le \gamma_S, \ \forall\, t \in [t_k, t_{k+N}) \tag{34e}$$

$$V(\tilde{x}(t_k)) \le \rho_e, \ \forall\, t \in [t_k, t_{k+N}), \ \text{if } \tilde{x}(t_k) \in \Omega_{\rho_e} \tag{34f}$$

$$\dot{V}(\tilde{x}(t_k), u) \le \dot{V}(\tilde{x}(t_k)), \Phi(\tilde{x}(t_k)), \ \text{if } \tilde{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_e} \tag{34g}$$

where $L_e(\tilde{x}(t), u(t))$ denotes the economic loss function, and $\tilde{x}(t)$ is predicted using the RNN model F_{nn}. In Eq. (34c), the decrypted plaintext $\hat{x}(t_k)$ is utilized to update the state value at t_k for the EMPC. Unlike Eq. (15), a new constraint is introduced in Eq. (34e) to limit the rate of change in the reference trajectory, where γ_S is a positive constant. Note that the solution of Eq. (34) is the input signal u, the optimal state trajectory x_E over the time period $[t_k.t_k + \Delta)$ is obtained using the predictive model in Eq. (34b) and the input signal is implemented in a sample-and-hold fashion. Since the optimal solution of the EMPC is transmitted to the PI controller, the limitation in Eq. (34e) ensures that the lower layer can track the obtained optimal state trajectory x_E. Eqs. (34f) and (34g) are the Lyapunov constraints to guarantee the closed-loop stability of the nonlinear system. It is worth noting that only linear mathematical operations are permissible in the encrypted space. Therefore, in the EMPC, the encrypted state signal is decrypted to the plaintext, which is utilized in the calculation. Then, the resulting optimal state set point is encrypted into the ciphertext e_2 and transmitted to the linear controller.

The PI controller in the lower layer approximated by the recursive rule is described as

$$u(t_k) = K_P e(t_k) + K_I e(t_k) + I_{t_{k-1}} \tag{35}$$

where K_P and K_I denote the proportional and integral gains, and $I_{t_{k-1}}$ is the integral control signal at t_{k-1}. $e(t_k) = \hat{x}_E(t_k) - \hat{x}(t_k)$ denotes the error between the encrypted state set point $\hat{x}_E(t_k)$ via LMPC and the encrypted state signal $\hat{x}(t_k)$ via sensor. The control signal generated by the PI controller is calculated through a linear operation on the encrypted information, remaining in the encrypted space. The control signal is transmitted to the actuator, where the encrypted input signal is decrypted and converted into the quantized input signal $\hat{u}(t_k)$, which is then applied to the nonlinear process.

Connections between the upper and lower tiers can be described as: (1) the ML-based EMPC in the upper tier calculates the set-point, and transmits the set-point to the lower tier; (2) the decryption is conducted and the plaintext is utilized in the calculation of the EMPC, and then the optimal solution obtained is encrypted. While in the lower tier, the encrypted information is provided to the PI controller, and the encrypted input signal is transmitted to the actuator. The two-layer encrypted ML-based EMPC scheme in Fig. 6 improves the closed-loop performance of the nonlinear system through the collaboration between the MPC and PI controller. The ML-based EMPC utilizes the machine learning model to predict the system dynamics for the calculation of the state set-point with optimal economic profit, and the PI controller is designed to track the optimal state trajectory. Since it is calculated in the encrypted space (i.e., linear operation of the encrypted state set-point from EMPC and the encrypted state signal from sensors), the lower layer with linear PI controllers improves the security, of which the encrypted control action is transmitted to the actuator. In addition, since only the actuator and EMPC have access to the private key for the decryption, the risk of cyber-attacks is reduced for the proposed two-layer framework. Finally, closed-loop stability can be guaranteed with selected parameters in the proposed scheme following the procedure outlined in Kadakia et al. (2024b).

5.2. Encrypted distributed ML-based MPC

In this subsection, the encrypted distributed ML-based LMPC scheme is developed for large-scale nonlinear processes. In practice, chemical processes are often interconnected, and the presence of multiple input and state variables makes controlling such complex nonlinear systems challenging. The distributed MPC method consists of multiple individual MPCs, each dedicated to a specific subsystem. These controllers communicate and coordinate with each other to ensure effective control of the overall system (Chen et al., 2020, 2021; Kadakia et al., 2024a).

Note that communication between individual controllers within the closed-loop system increases the risk of cyber-attacks. To mitigate this risk, the proposed encrypted method is employed, ensuring secure information exchange while maintaining control performance and system stability. Additionally, the complexity of system dynamics makes it challenging to derive an explicit first-principles model. To address this, a data-driven neural network model is utilized as the predictive model in the distributed MPC scheme.

A nonlinear process network with two interconnected subsystems is used as an example to show the encrypted distributed ML-based LMPC scheme. The selection of two subsystems is for the simplicity of notation, and the proposed method can be readily extended to the case with multiple nonlinear subsystems. The proposed encrypted distributed ML-based LMPC scheme is illustrated in Fig. 7. Specifically, the implementation strategy for the encrypted control scheme is as follows:

1. At each sampling time t_k, the state signals $x(t_k)$ from the two subsystems are collected by the sensors, and then encrypted into the ciphertext e_x, which is transmitted to the individual controllers.
2. The ciphertext e_x is first decrypted to the plaintext $\hat{x}(t_k)$ in each individual controller, where it is used in the calculation of the ML-based LMPC.
3. At iteration $k = 1$, the b_{th} (i.e., $b = 1, 2$) LMPC evaluates the optimal trajectories of the control action $u_b(t)$ for $t \in [t_k, t_{k+N})$ using the quantized state $\hat{x}(t_k)$. For the calculation of b_{th} LMPC, the neural network model is utilized to predict the state trajectory to optimize the control actions. Moreover, it is assumed that the control action for the a_{th} subsystem is the neural-network-based controller $\boldsymbol{\Phi}^a_{nn}(\hat{x}(t_k))$, where $a = 1, 2$ and $a \neq b$.
4. After the calculation of the first iteration, the control action obtained by the b_{th} LMPC for the next sampling period is encrypted and transmitted to the actuator of the b_{th} subsystem. In addition, the computed control inputs over N sampling times for the two LMPCs are encrypted and exchanged between each other. For example, the control inputs u_1 obtained by the ML-based LMPC 1 over $[t_k, t_{k+N})$ is encrypted to the ciphertext e_1, and transmitted to the LMPC 2, where they are decrypted to the quantized plaintext $\hat{u}_1$.
5. At iteration $k > 1$:
   (a) Each LMPC recalculates the optimal future input trajectory for its own subsystem, based on the decrypted state measurement $\hat{x}$ and the quantized control input $\hat{u}$ of the other subsystems.
   (b) Subsequently, the new input trajectory is shared with the other LMPC following the encryption and decryption process in Step 4. The cost function is calculated and the value is stored.
   (c) A termination condition is set for the iterative distributed LMPC scheme. If the condition is met, the optimal input trajectory for the next sampling time is encrypted in each LMPC, and the ciphertext is transmitted to the corresponding subsystem. If the termination condition is not satisfied, reiterate the aforementioned steps ($5.a \rightarrow 5.b$).
6. The encrypted control action is decrypted at the actuator, and the quantized input action $\hat{u}_b(t)$ is implemented at the b_{th} subsystem over one sampling time period.

In Fig. 7, the communication between the sensors, the controllers, and the actuators is encrypted, developing a secure information exchange scheme for the distributed ML-based MPC. Specifically, the state information exchanged between the sensor and the controller is encrypted, and the control action is encrypted by the individual controller before being transmitted to each subsystem. Moreover, the control input trajectory, which is exchanged between the LMPCs in the iterative distributed scheme, is encrypted and then transmitted to other individual controllers, where the signal is decrypted using the private key.

The optimization problem of the b_{th} LMPC at iteration $k = 1$ is constructed as follows:

$$\mathcal{J} = \min_{u_b \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), \boldsymbol{\Phi}^a_{nn}(\tilde{x}(t)), u_b(t))\, \mathrm{d}t, \tag{36a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), \boldsymbol{\Phi}^a_{nn}(\tilde{x}(t)), u_b(t)) \tag{36b}$$

$$u_b(t) \in U_B, \ \forall\ t \in [t_k, t_{k+N}) \tag{36c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{36d}$$

$$\dot{V}(\hat{x}(t_k), \boldsymbol{\Phi}^a_{nn}(\hat{x}(t_k)), u_b(t_k)) \leq \dot{V}(\hat{x}(t_k), \boldsymbol{\Phi}^a_{nn}(\hat{x}(t_k)), \boldsymbol{\Phi}^b_{nn}(\hat{x}(t_k))), \quad \text{if } \hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{36e}$$

$$V(\tilde{x}(t)) \leq \rho_{\min}, \ \forall\ t \in [t_k, t_{k+N}), \quad \text{if } \hat{x}(t_k) \in \Omega_{\rho_{\min}} \tag{36f}$$

The system dynamics is predicted using the neural network in Eq. (36b). In addition, the neural network-based candidate controller $\boldsymbol{\Phi}_{nn}$ is selected for the calculation of the LMPC. At iteration $k > 1$, following the exchange of the optimal input trajectories $\hat{u}_a(t)$, the b_{th} LMPC is modified as:

$$\mathcal{J} = \min_{u_b \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), \hat{u}_a(t), u_b(t))\, \mathrm{d}t, \tag{37a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), \hat{u}_a(t), u_b(t)) \tag{37b}$$

$$u_b(t) \in U_b, \ \forall\ t \in [t_k, t_{k+N}) \tag{37c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{37d}$$

$$\dot{V}(\hat{x}(t_k), \hat{u}_a(t_k), u_b(t_k)) \leq \dot{V}(\hat{x}(t_k), \boldsymbol{\Phi}^a_{nn}(\hat{x}(t_k)), \boldsymbol{\Phi}^b_{nn}(\hat{x}(t_k))), \quad \text{if } \hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{\min}} \tag{37e}$$

$$V(\tilde{x}(t)) \leq \rho_{\min}, \ \forall\ t \in [t_k, t_{k+N}), \quad \text{if } \hat{x}(t_k) \in \Omega_{\rho_{\min}} \tag{37f}$$

The key difference for the b_{th} LMPC at iterations $k = 1$ and $k > 1$ lies in the calculation of $\dot{V}$. For the first iteration, the control action for the a_{th} subsystem is estimated using the NN-based candidate controller $\boldsymbol{\Phi}^a_{nn}(\hat{x}(t_k))$. At iteration $k > 1$, the time derivative of the Lyapunov function in Eq. (37e) is calculated using the decrypted control signal $\hat{u}_a(t_k)$, which is exchanged from the a_{th} LMPC. The exchange of control actions between each individual controller in the iterative distributed control scheme can improve the closed-loop performance, since the updated optimal control actions from other controllers are utilized in the calculation of the ML-based LMPC. However, the frequent communication between individual controllers increases the risk of cyber-attacks. In this work, despite the encrypted information transformation between the sensor and the controller, as well as the controller and the actuator, the optimal control trajectory is encrypted and exchanged between individual controllers, which enhances the cybersecurity of the closed-loop system.

6. Conclusion

This work presented a machine learning-based model predictive control scheme that uses encryption in the measurement-controller and controller-actuator links to improve cybersecurity. Specifically, to improve cybersecurity, the measurement signals and the control signals that the MPC employed and calculated, respectively, were encrypted using the Pallier cryptosystem with varying degrees of precision. The impact of encryption quantization error on closed-loop stability was studied. A model predictive control system that utilized a machine learning model to describe the process dynamics was used to evaluate the impact of encryption error in the context of a nonlinear non-isothermal continuous stirred tank reactor example. The data loss from encryption induced quantization error led to minor deviations from the baseline (without any encryption) closed-loop state trajectory with

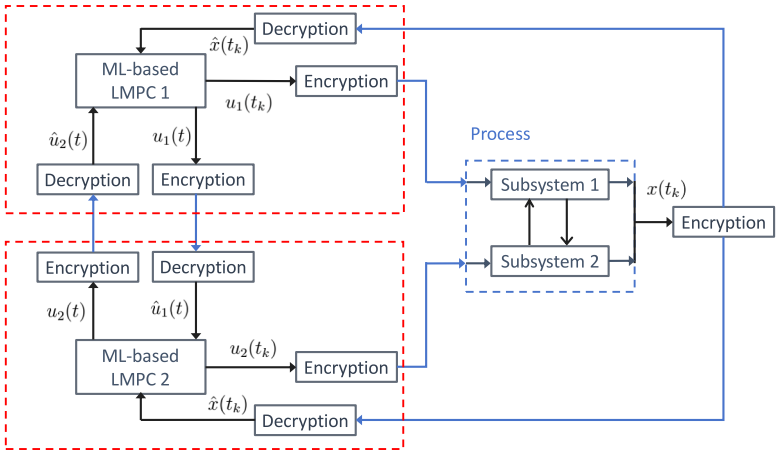

Fig. 7. Encrypted distributed ML-based MPC.

more noticeable deviations present in the resulting control inputs that scaled inversely with the size of the precision parameter d. The simulation results demonstrated how encryption can be used in a chemical process application with minimal impact on system performance while simultaneously increasing security against cyber-attacks. Finally, the encrypted ML-based MPC framework was extended to a two-layer economic model predictive control framework and a distributed model predictive control scheme.

CRediT authorship contribution statement

Arthur Khodaverdian: Writing – original draft, Methodology, Investigation, Conceptualization. **Guoquan Wu:** Writing – original draft, Methodology, Investigation, Conceptualization. **Zhe Wu:** Writing – original draft, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization. **Panagiotis D. Christofides:** Writing – original draft, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Financial support from National Science Foundation, Department of Energy, and NRF-CRP 27-2021-0001, Singapore is gratefully acknowledged.

Data availability

Data will be made available on request.

References

Arauz, T., Chanfreut, P., Maestre, J., 2022. Cyber-security in networked and distributed model predictive control. Annu. Rev. Control. 53, 338–355.
Benny, S., Desai, I., Uriarte, L., Tsai, I., McMahan, L., 2024. A meta-analysis on NIST post-quantum cryptographic primitive finalists. J. Emerg. Investig. 7.
Chen, S., Wu, Z., Christofides, P.D., 2021. Cyber-security of centralized, decentralized, and distributed control-detector architectures for nonlinear processes. Chem. Eng. Res. Des. 165, 25–39.
Chen, S., Wu, Z., Rincon, D., Christofides, P.D., 2020. Machine learning-based distributed model predictive control of nonlinear processes. AIChE J. 66, e17013.
Darup, M.S., 2020. Encrypted model predictive control in the cloud. In: Privacy in Dynamical Systems. Springer Singapore, pp. 231–265.
Darup, M.S., Redder, A., Shames, I., Farokhi, F., Quevedo, D., 2017. Towards encrypted MPC for linear constrained systems. IEEE Control. Syst. Lett. 2, 195–200.
Data61, C., 2013. Python paillier library. https://github.com/data61/python-paillier. (Accessed 12 August 2024).
Hassanpour, H., Corbett, B., Mhaskar, P., 2020. Integrating dynamic neural network models with principal component analysis for adaptive model predictive control. Chem. Eng. Res. Des. 161, 26–37.
Intel, 2024. Intel®64 IA-32 architectures software developer's manual combined volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4. Section 4.2.1.2.
Kadakia, Y.A., Abdullah, F., Alnajdi, A., Christofides, P.D., 2024a. Encrypted distributed model predictive control of nonlinear processes. Control Eng. Pract. 145, 105874.
Kadakia, Y.A., Abdullah, F., Alnajdi, A., Christofides, P.D., 2024b. Integrating dynamic economic optimization and encrypted control for cyber-resilient operation of nonlinear processes. AIChE J. 70 (9), e18509.
Khalil, H.K., 2002. Nonlinear Systems. Prentice Hall.
Limon, D., Calliess, J., Maciejowski, J.M., 2017. Learning-based nonlinear model predictive control. IFAC- Pap. 50, 7769–7776.
National Institute of Standards and Technology, 2024. The NIST Cybersecurity Framework (CSF) 2.0. Technical Report, National Institute of Standards and Technology.
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. Springer, Berlin, Heidelberg, pp. 223–238.
Parker, S., Wu, Z., Christofides, P.D., 2023. Cybersecurity in process control, operations, and supply chain. Comput. Chem. Eng. 171, 108169.
Schlüter, N., Binfet, P., Darup, M.S., 2023. A brief survey on encrypted control: From the first to the second generation and beyond. Annu. Rev. Control. 56, 100913.
Stobbe, P., Keijzer, T., Ferrari, R.M., 2022. A fully homomorphic encryption scheme for real-time safe control. In: Proceedings of 61st IEEE Conference on Decision and Control. Cancun, Mexico, pp. 2911–2916.
Sui, T., Wang, J., Liu, W., Pan, J., Wang, L., Zhao, Y., Kong, L., 2024. Optimizing encrypted control algorithms for real-time secure control. J. Franklin Inst. 361 (5), 106677.
Suryavanshi, A., Alnajdi, A., Alhajeri, M., Abdullah, F., Christofides, P.D., 2023. Encrypted model predictive control design for security to cyberattacks. AIChE J. 69, e18104.
Terzi, E., Bonassi, F., Farina, M., Scattolini, R., 2021. Learning model predictive control with long short-term memory networks. Internat. J. Robust Nonlinear Control 31, 8877–8896.
Wong, W., Chee, E., Li, J., Wang, X., 2018. Recurrent neural network-based model predictive control for continuous pharmaceutical manufacturing. Mathematics 6, 242.
Wu, Z., Christofides, P.D., Wu, W., Wang, Y., Abdullah, F., Alnajdi, A., Kadakia, Y., 2025. A tutorial review of machine learning-based model predictive control methods. Rev. Chem. Eng. 41, 42.
Wu, Z., Tran, A., Rincon, D., Christofides, P.D., 2019. Machine-learning-based predictive control of nonlinear processes. Part II: Computational implementation. AIChE J. 65 (11), e16734.
Xu, Z., Wu, Z., 2024. Privacy-preserving federated machine learning modeling and predictive control of heterogeneous nonlinear systems. Comput. Chem. Eng. 187, 108749.