Digital Chemical Engineering 15 (2025) 100233

Contents lists available at ScienceDirect

Digital Chemical Engineering

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

Original article

Enhancing cybersecurity of nonlinear processes via a two-layer control architecture

Arthur Khodaverdian [a], Dhruv Gohil [a], Panagiotis D. Christofides [a,b,*]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095, USA
[b] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA

ARTICLE INFO

Keywords:
Encrypted control
Model predictive control
Paillier encryption
Nonlinear processes
Semi-homomorphic encryption

ABSTRACT

This work proposes a novel two-layer multi-key control architecture to enhance the resilience of nonlinear chemical processes to cyberattacks. The architecture consists of an upper-layer nonlinear controller and a lower-layer of encrypted linear controllers. The nonlinear controllers process unencrypted sensor data to determine optimal control actions, which are then used to estimate the closed-loop state trajectory using a first-principle model of the plant. This trajectory is sampled and mapped to a valid subset before encryption, which can lead to minor inaccuracies. The resulting encrypted state-space data samples are used as set-points for the lower-layer controllers, which can be implemented using encrypted signals, allowing for obfuscation of the computation and transmission of the applied control inputs, thereby enhancing cybersecurity. This study further improves security by taking advantage of the Single-Input-Single-Output nature of some linear control methods to allocate a unique encryption key to each linear controller and its respective sensor data. Two nonlinear chemical process applications, including a benchmark chemical reactor example and one application modeled through the use of Aspen Dynamics, are used to demonstrate the application of the proposed two-layer architecture.

1. Introduction

Supervisory Control and Data Acquisition (SCADA) technology, an integral aspect of Industrial Control Systems (ICS), has revolutionized the management of complex operations by streamlining the transmission of data via networked communications. Although this development has simplified the managerial aspect of information transfer, it has also introduced new modes of compromising operations in the form of cybersecurity vulnerabilities. The networked communications are susceptible to various cyberattacks, such as signal manipulation, that can result in financial loss, infrastructure failure, and even loss of life (Babu et al., 2017; Gandhi et al., 2011; Simmons et al., 2009). Networked communications are a core aspect of Information Technology (IT), which involves the storage, transmission, and processing of data, but despite the inherent susceptibility to cyber vulnerabilities, IT's substantial progress in cybersecurity measures has minimized the frequency and magnitude of these attacks ((ACSC), 2024). Industrial operations are more akin to Operational Technology (OT) systems, which, unlike IT systems, have lagged behind in cybersecurity, necessitating the development of robust security measures tailored to the unique challenges of control systems (Conklin, 2016). The consequence of this negligence has already resulted in an increase in malicious attacks. In recent history, cyberattacks have been used to cripple critical infrastructure worldwide and are increasingly being used in warfare: for example, in 2010, a cyberworm dubbed Stuxnet damaged centrifuges used by Iran for uranium enrichment (Farwell and Rohozinski, 2011); on September 26, 2024, Microsoft reported findings on a ransomware-as-a-service organization dubbed Storm-0501 that exploited weak security to launch widespread ransomware attacks that even managed to infiltrate cloud environments (Microsoft Security Team, 2024). To prevent future cyberattacks, it is necessary to improve the cybersecurity of OT infrastructures. Cybersecurity, particularly for large-scale processes, has been a key focus in recent studies. From this, a wide range of methodologies have been devised, each with their own trade-offs.

An increasingly popular tool in the field of automatic control is machine learning, or more specific to this context, machine learning based cyberattack detectors (Wang et al., 2022; Aljohani et al., 2024). Sadly, the progression in cybersecurity has been met with equivalent progression in cyberattacks. Research into machine learning tools have uncovered a unique vulnerability in the form of adversarial attacks, and despite the progress in developing countermeasures to these types of cyberattacks, the issues of limited data, particularly on attack methods,

* 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.dche.2025.100233
Received 28 February 2025; Received in revised form 21 March 2025; Accepted 21 March 2025
Available online 1 April 2025

has complicated the development of these tools (Koay et al., 2023; Chen et al., 2017).

Other promising alternatives included miscellaneous backup control schemes (Huang et al., 2022), post-cyberattack system recovery (Wu et al., 2020), or more recently, self-healing and fault-tolerant systems (Aldrini et al., 2024). These methods focus on mitigating or recovering from successful cyberattacks, and are powerful tools that should be incorporated if possible; however, such methods should not be used as an excuse to neglect preventative measures. 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 cyberattacks. By employing encryption schemes (e.g., homomorphic encryption), these systems allow for secure transmission of data and control actions over potentially vulnerable networks.

Research involving the prevention of cyberattacks in the first place by means of linear encrypted controllers (Pan et al., 2023b,a) or otherwise cyberattack-resilient controllers (Paridari et al., 2017) achieves the desired goal of cyber-resilience, but generally comes at the consequence of requiring simple linear control schemes. Linear schemes have the benefit of being able to utilize semi-homomorphic encryption, which allows for specific operations such as addition and multiplication to be applied to encrypted data without the need to decrypt. Relative to nonlinear alternatives such as model predictive control (MPC), these control schemes perform poorly and with worse cost-efficiency, especially when applied to systems with highly nonlinear dynamics (Geng and Yang, 2014). Although homomorphic encryption allows specific operations (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). Thus, recent studies have considered the use of both advanced nonlinear control schemes such as Lyapunov-based economic model prediction control (LEMPC) and traditional linear control schemes such as proportional integral (PI) control to promote performance while allowing for improved cybersecurity (Kadakia et al., 2024).

Building on this development, we propose, in the present work, an encrypted two-layer control architecture that utilizes multiple encryption keys. In this architecture, the upper-layer is composed of a nonlinear controller that reads sensor data from the plant and generates control inputs that would optimize the economics of the plant over some future trajectory. Due to the nonlinearity of these controllers, calculations are done in plaintext, which introduces a vulnerability. To mitigate this, the optimal control inputs are not directly applied to the process. Instead, a first-principles model of the plant is used to estimate the process state-trajectory from the optimal control inputs. This trajectory is sampled and used as set-points for the lower-layer control system to utilize. The lower-layer control is made of distributed linear controllers, each utilizing their own encryption key, which track the provided set-points by running calculations using only encrypted signals. In order to solve arithmetic equations while encrypted, a homomorphic encryption scheme needs to be used. Homomorphic encryption schemes allow for addition and multiplication of encrypted data, which enables the use of linear encrypted controllers. Using a classical nonlinear chemical process example, this two-layer control scheme is first demonstrated to allow for secure control of a nonlinear system with minimal losses in performance. Subsequently, to further demonstrate the viability of this method in a large-scale case study, the two-layer control architecture is applied to a chemical process application modeled through the use of Aspen Dynamics, to demonstrate the industrial viability of the approach.

2. Preliminaries

2.1. Notation

The transpose of vector x is denoted by $x^\top$. The set of real numbers, integers, and natural numbers are denoted by $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$, respectively. $\mathbb{Z}_M$ denotes the additive groups of integers modulo M while $\mathbb{Z}_M^*$ denotes the multiplicative groups of integers modulo M. Set subtraction of set B from set A yields a set of elements that are in set A but not in set B and is denoted as $A \backslash B$. Functions are denoted as $f(\cdot)$. $\mathrm{lcm}(i,j)$ and $\gcd(i,j)$ denote the least common multiple and greatest common divisor of integers i and j, respectively. $x//y$ denotes Integer Division as implemented in Python, which is functionally identical to the floor operation applied to the result of regular division, denoted $\lfloor \frac{x}{y} \rfloor$.

2.2. Class of systems

This work focuses on nonlinear multiple-input multiple-output (MIMO) continuous-time systems described by nonlinear first-order ordinary differential equations (ODE) of the form shown below:

$$\dot{x} = F(x,u) = f(x) + \sum_{i=1}^{m} g_i(x) u_i \tag{1}$$

The state vector $x = [x_1, x_2, \dots, x_n] \in \mathbb{R}^n$ describes the raw sensor readings, the input vector $u = [u_1, u_2, \dots, u_m] \in \mathbb{R}^m$ describes the applied control inputs where $m \leq n$. Each control input is bounded, $u_{i,\min} \leq u_i \leq u_{i,\max} \forall i = 1, 2, \cdots, m$ where $u_{i,\min}$ and $u_{i,\max}$ represent the lower and upper bounds of each control action, respectively. The functions $f(\cdot)$ and $g_i(\cdot) \forall i = 1, 2, \cdots, m$ are assumed to be sufficiently smooth vector functions. Without loss of generality, we treat the origin as a steady-state of Eq. (1) by assuming that $f(0) = 0$ (or $F(0,0) = 0$). We further designate the initial time as zero ($t_0 = 0$). Finally, the set $S(\Delta)$ is defined as the assortment of piece-wise constant functions characterized by a period of Δ.

2.3. Paillier cryptosystem

In order to apply encryption to a linear process, homomorphic encryption needs to be applied. Namely, additively semi-homomorphic encryption allows for the error terms present in traditional PI control to be encrypted while leaving the gains as encoded weights. Among the various options, the Paillier cryptographic system was chosen due to it being a probabilistic additively semi-homomorphic asymmetric encryption scheme that simultaneously achieves low encryption cost and a low value of expansion relative to other homomorphic options, although any additively homomorphic scheme with support for an arbitrary number of operations would work for this design (Sen, 2013). Asymmetric methods are particularly useful in the context of the two layer scheme, as it allows for the exposure of public-keys in the upper-layer without the risk of directly compromising the lower-tier controllers. Additionally, the added time complexity relative to symmetric methods results in a conservative approach for determining the impact of encryption time on the process, as excessive time use

during these computations increases the required sampling time of the system, which worsens the ability to control the process.

The Paillier cryptosystem (Paillier, 1999) is used to encrypt raw sensor readings (x) and calculated set-points (x_{sp}). Control inputs calculated by the linear controllers (u) using these encrypted readings will be decrypted locally at their relevant equipment. Due to the Paillier cryptosystems additively semi-homomorphic nature, we can solve a subset of linear calculations without decrypting sensor data. Each linear controller will be designated its own unique pair of public and private keys for encryption. Sensor readings and set-points will be converted into positive integers prior to encryption. The linear controllers will proceed to run calculations on encrypted data without needing to decrypt intermediary results, ensuring security. The data is encrypted using the controller's respective public key, and decrypted using the controller's respective private key. Both keys are generated as follows:

1. Randomly generate two large prime integers (p and q).
2. Check if $\gcd(pq,(p-1)(q-1)) = 1$. If true, proceed, otherwise, repeat 1.
3. Solve $M = pq$.
4. Generate an integer $g \in \mathbb{Z}^*_{M^2}$.
5. Solve $\lambda = \mathrm{lcm}(q-1, p-1)$.
6. Define the function $\bar{L}(x) = (x-1)/M$.
7. Check if the modular multiplicative inverse exists: $u = (\bar{L}(g^\lambda \bmod M^2))^{-1} \bmod M$. If not, repeat from (4). If it exists, (M, g) is the public key and (λ, u) is the private key.

This key-generation process is done for every linear controller in the plant, and the keys are securely stored and distributed. The encryption formula is as follows:

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

such that $r \in \mathbb{Z}_M$ and is randomly generated. The resulting ciphertext form of m is denoted as c. To decrypt, the equation is as follows:

$$D_M(c) = m = \bar{L}(c^\lambda \bmod M^2)u \bmod M \tag{3}$$

2.4. Quantization and mapping

The Paillier cryptosystem only operates on non-negative integers, thus, data must be mapped from $\mathbb{R}$ to $\mathbb{Z}_M$ by means of quantization and subsequent bijective mapping (Schulze Darup et al., 2018). Quantization involves mapping from $\mathbb{R}$ to $\mathbb{Q}_{l,d}$ where $\mathbb{Q}_{l,d}$ is the set of signed fixed-point binary numbers of bit length l and fractional bit length d. Using the 2's complement representation (Intel, 2024) the set can be defined as $\mathbb{Q}_{l,d} = \{q \in \mathbb{Q} | q = -2^{l-d-1}\beta_l + \sum_{i=1}^{l-1} 2^{i-d-1}\beta_i, \beta_i \in \{0,1\} \forall i = 1 \cdots l\}$. For $a \in \mathbb{R}$, we define the function $g_{l,d}$:

$$\begin{aligned} g_{l,d} &: \mathbb{R} \to \mathbb{Q}_{l,d} \\ g_{l,d}(a) &:= \arg\min_{q \in \mathbb{Q}_{l,d}} |a - q| \end{aligned} \tag{4}$$

Bijective mapping of the quantized data to the set of non-negative integers is denoted by $f_{M,d}$, as outlined in Schulze Darup et al. (2018), to ensure that the quantized data is mapped into a subset of the message space $\mathbb{Z}_M$:

$$\begin{aligned} f_{M,d} &: \mathbb{Q}_{l,d} \to \mathbb{Z}_M \\ f_{M,d}(q) &:= 2^d q \bmod M \end{aligned} \tag{5}$$

To complete the encryption process, elements in this space are converted to ciphertexts using Eq. (2) which can then be decrypted back to the message space. The following inverse mapping, denoted as $f^{-1}_{M,d}$, allows for the retrieval of the quantized form:

$$f^{-1}_{M,d} : \mathbb{Z}_M \to \mathbb{Q}_{l,d} \tag{6}$$

$$f^{-1}_{M,d}(m) := \frac{1}{2^d} \begin{cases} m - M & \text{if } m \geq M - M//3 + 1 \\ m & \text{otherwise} \end{cases} \tag{7}$$

In this particular implementation of Pallier encryption, ~1/3 of the message space is allocated to detecting overflow. We can define the positive quadrant as the first $M//3$ items (i.e. $m < M//3$ or $m \leq M//3 - 1$) and the negative quadrant as the last $M//3$ items (i.e. $m > M - M//3$ or $m \geq M - (M//3 - 1)$). The remaining values where $M//3 \leq m \leq M - M//3$ are considered invalid values. Any number that encodes into this region or decodes from this region are considered invalid or the result of overflow respectively.

2.5. Encrypted arithmetic

As an additively semi-homomorphic encryption scheme, Paillier encryption allows for an arbitrary number of arithmetic operations on ciphertexts. In particular, plaintext addition is achieved through the multiplication of the corresponding ciphertexts, as shown below.

$$\begin{aligned} &D_M((E_M(m_1,r_1) \cdot E_M(m_2,r_2)) \bmod M^2) \\ &= (m_1 + m_2) \bmod M \end{aligned} \tag{8}$$

For this to work, any value m, given the same r, must encrypt to the same ciphertext. Because of the quantization and bijective mapping steps, depending on the fractional bit length, the same number may map to different values within the message space. To prevent this, d is tracked. When two terms with different values of d are added, the term with the larger d value will be modified to share the smaller d value using the equation below:

$$m_{d_2} = m_{d_1} 2^{d_1 - d_2} \; : \; d_1 < d_2 \tag{9}$$

Plaintext multiplication on the other hand is achieved by:

$$D_M(E_M(m_1,r_1)^k \bmod M^2) = k m_1 \bmod M \tag{10}$$

Note that in the case of multiplication, the exponent is plaintext, which means multiplication is not possible while keeping both terms encrypted. Further note that the multiplication process will lead to a new value for d equal to the sum of the d values of the terms being multiplied.

2.6. Overflow

In the process of quantizing and bijectively mapping real valued data to a subset of non-negative integers, we introduced a scaling factor of 2^d. This scaling factor can change as arithmetic operations are done. Addition will result in the highest precision of the two, and Multiplication will result in a new precision exponent equal to the sum of the precision exponents (i.e $2^{d_1}2^{d_2} = 2^{d_1+d_2} = 2^{d_3}$); however, the size of the message space is a constant that is predefined during the key generation process. Due to modulus operations, negative valued data will map to the upper half of the message space. Because of this mapping, there is a sharp transition between the positive and negative halves of the message space that data can cross. This is called overflow, and it may ruin data if not caught. To help in detection of overflow, the message space is segmented into 3 partitions. Positive real numbers are mapped to the lower third of this space, while negative numbers are mapped to the upper third. The center is left empty, meaning that entering this zone is only possible from overflow. This is done by enforcing a maximum value of ~1/3 of the message space, which is seen in the reverse bijective mapping in Eq. (7).

3. Development of the two-layer multi-key encrypted control architecture

This section will cover the design of the proposed two-layer multi-key encrypted control architecture and its corresponding nonlinear and linear controllers.

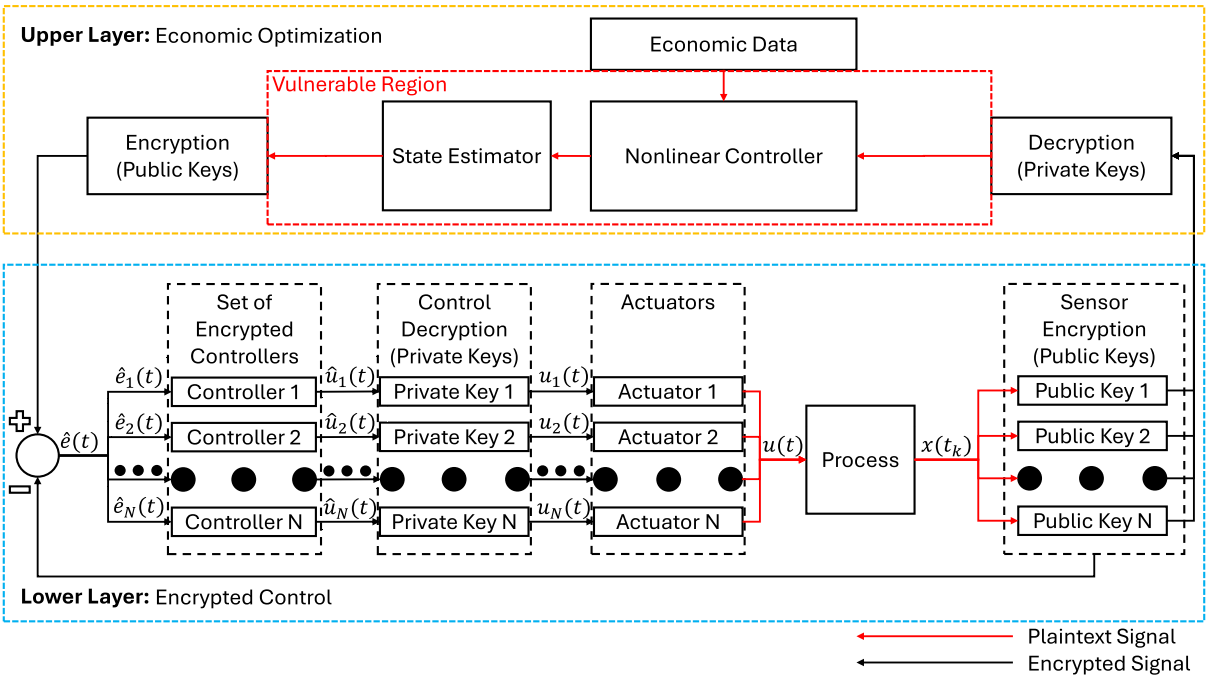

Fig. 1. Two-layer control architecture block diagram.

3.1. Design and implementation

As illustrated in Fig. 1, the reference time frame is t_k. All sensor signals $x(t_k)$ are encrypted using a unique public key and become ciphertext. These signals are sent to a nonlinear controller where they are then decrypted. The controller solves the optimal control inputs for some time horizon. These inputs are not directly implemented, instead, they are passed to a first-principles model that calculates an estimate of the state-trajectory. This trajectory is sampled, and the sampled state values are encrypted with their respective public key before being transmitted to the linear controllers for computation of the applied control input. Encrypted sensor signals are also sent to the linear controllers to compute the current error with respect to the current set-point. By using the Paillier cryptosystem, the error can be calculated without decrypting the ciphertext; however, PI controller weights or other constant parameter multiplications are left as plaintext, which leaves them vulnerable. The encrypted control actions are transmitted to their respective actuators, at which point the signal is decrypted. In the case of model-predictive-control (MPC), this continues until all set-points corresponding to the first MPC control action are used, at which point the MPC recalculates the optimal control inputs.

As seen in the system design shown in Fig. 1, there are four primary vulnerable points to cyberattacks: the nonlinear controller's decrypted state input and unencrypted optimal control signals, the unencrypted set-points, and economic data used in the nonlinear controller. Cyberattack detection can be implemented to minimize the risks posed by these components, but in this case, security of the encrypted signals is prioritized (Paridari et al., 2017). To minimize the risk of the entire system being compromised in the case of private/public keys being deciphered, we have implemented multiple keys, where one key corresponds to one linear controller. If this was not the case, the keys being compromised would compromise the entire process, whereas multiple keys isolate the damage to individual units, allowing for additional countermeasures. Additionally, this allows for each linear controller to individually verify that the signal received uses the correct encryption key. If not, a simple backup control design would be to default to steady-state operation for safety until a new key gets implemented.

Because Pallier encryption is used, this design introduces quantization error to the state variable readings, set-point calculations, and control input calculations with a magnitude less than 2^{-d}. An additional benefit of the multi-key design is the ability to vary this quantization error as needed for each linear controller. Lower precision can be used for larger terms such as the heat input to allow for improved overflow detection, whereas higher precision can be used for terms that require more precise numerical estimations, such as concentration measurements.

Remark 1. The upper-layer shown in this paper does not necessarily need to be an MPC or EMPC. The purpose of the upper-layer is to provide economically optimized set-points for the lower-layer to track. Thus, the upper-layer can be of any type of nonlinear controller, so long as it provides set-points to the lower-layer.

Remark 2. Although there are still cybersecurity vulnerabilities present in the proposed control architecture, because no component has both the public and private key simultaneously, and all signals relevant to the control signal calculation are encrypted, the risks of cyberattacks are limited to cases where components are physically compromised or cases where keys are deciphered.

3.2. Dynamic economic optimization

The upper-layer controller can be formulated as a Lyapunov-based (economic) model-predictive-controller (LEMPC) formulated as follows:

$$\mathcal{J} = \max_{u \in S(\Delta)} \int_{t_k}^{t_k+N\Delta} L\left(\tilde{x}(t), u(t)\right)\, \mathrm{d}t \tag{11a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t), u(t)) \tag{11b}$$

$$u \in U, \ \forall\, t \in [t_k, t_k + N\Delta) \tag{11c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{11d}$$

$$\left.\begin{aligned} &\text{if } \tilde{x}(t_k) \in \Omega_{\rho_{secure}}, \ \forall\, t \in [t_k, t_k + N\Delta) \\ &V(\tilde{x}(t_k)) \le \rho_{secure} \end{aligned}\right\} \tag{11e}$$

$$\left.\begin{aligned} &\text{if } \tilde{x}(t_k) \in \Omega_{\rho} \backslash \Omega_{\rho_{secure}}, \\ &\dot{V}(\tilde{x}(t_k), u) \le \dot{V}(\tilde{x}(t_k), \Phi(\tilde{x}(t_k))) \end{aligned}\right\} \tag{11f}$$

In this formulation, Δ is the sampling period representing the duration of the control input u as well as the time between sensor readings of the state x. Eq. (11a) is the cost function that must be optimized, which incorporates economics using a function of both the predicted state values $\tilde{x}$ and the current optimal determination for the control input trajectory over the prediction horizon, u. Eq. (11b) represents the first-principles model that is used to simulate system dynamics, where Eq. (11d) initializes the state measurements of the model using quantized sensor readings. (t_k) denotes an arbitrary initial time-frame, whereas N denotes the total number of sampling periods that the LEMPC will optimize the control inputs for. The remaining constraints are necessary to ensure stability of the closed-loop system by utilizing a Lyapunov function, the mathematical proof of which is discussed in our prior work (Kadakia et al., 2024). Eq. (11e) ensures that if the current state values lie within an economic region represented by a level set of the Lyapunov function, then it will remain in this level set for all future times. As disturbances, delays, and dynamics can lead to a violation of this in future sensor readings, we include a second operating mode represented by Eq. (11f) where the rate of change of the Lyapunov function must be more negative than the rate induced by a stabilizing reference control that is assumed to exist.

3.3. Constraining the rate of change of control actions

PI control is not traditionally implemented in a manner where the set-points change with time, but in this implementation, the sudden set-point change, especially in the early stages of the process, will result in large jumps in the error term. To improve the feasibility of the control action, we clip the control action proportionally to the decision variables magnitude. Specifically, any change in the heating terms is bounded by a maximum change in magnitude of $\pm 0.2T^2$. Alternative forms of this can be chosen as needed, but in order to remain secure while applying this, it is important that this step be done either with a secure offline module to allow for safe decryption, or it can be done using fully-homomorphic encryption with some approximation of the min/max operations.

3.4. Constraining the rate of change of set-point update

An alternative to the constraint shown in Section 3.3 is to improve the linear controllers ability to track the set-points generated by simulating and sampling the closed-loop state trajectory under the optimal control actions calculated by the LEMPC by enforcing a maximum rate of change constraint to the set-points sent to the lower-layer over two consecutive set-points. This method, along with the method in Section 3.3 are done as alternatives to enforcing additional constraints to the LEMPC system above, as additional constraints increase computational time and may worsen an optimization tools ability to find solutions; specific details for the implementation of these constraint are given in the application section below.

4. Application to a benchmark chemical process example

To demonstrate the proposed encrypted control framework, we will first consider a benchmark nonlinear chemical process example. The objective is to operate the process around a steady-state and improve process economic behavior for approaching a desired operating point relative to traditional PI control and tracking MPC to demonstrate the frameworks economic benefits over the former, and minimal loss of performance compared to the latter. Solving the optimization problem of the upper-layer LEMPC requires a nonlinear solver. In this example, we selected to use a Python implementation of IPOPT via the Cyipopt module (mechmotum, 2018).

Table 1
Parameter values for the chemical process example.

Name	*Label*	*Value*	*Units*
Flow Rate	F	5	m^3/hr
Reactor Volume	V	1	m^3
Pre-exponential Factor	k_0	8.46×10^6	$m^3/(kmol\,hr)$
Activation Energy	E	5×10^4	kJ/kmol
Gas Constant	R	8.314	kJ/(kmol K)
Liquid Density	ρ_L	1000	kg/m^3
Enthalpy of Reaction	ΔH	-1.15×10^4	kJ/kmol
Inlet Temperature	T_0	300	K
s.s Heat Input Rate	$\dot{Q}_s$	0	kJ/hr
s.s Inlet Concentration	C_{A0_s}	4	$kmol/m^3$
s.s Concentration	C_{As}	1.9537	$kmol/m^3$
s.s Temperature	T_s	401.87	K
Specific Heat	C_p	0.231	kJ/(kg K)

s.s stands for steady-state

4.1. Process model and control problem

A perfectly mixed continuous stirred tank reactor (CSTR) is used to convert reactant A to product B by means of a second-order irreversible exothermic elementary reaction $A \rightarrow B$. The temperature of the CSTR is regulated using a jacket. For simplicity, the heat transfer dynamics of a jacket are reduced to a direct heat input term ($\dot{Q}$) which can be negative or positive as needed. A second-order rate law is used for the reaction in question, i.e., $r_A = k_0 e^{-\frac{E}{RT}} C_A^2$. Using dynamic mass and energy balances, we can derive the following dynamic model of the process, consisting of two first-order ODEs:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 \tag{12a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{\dot{Q}}{\rho_L C_p V} \tag{12b}$$

The constant parameters used in these ODEs are defined in detail in Table 1. The reactor concentration (C_A) and temperature (T) are chosen to be the state variables, whereas the heat input rate ($\dot{Q}$) and inlet concentration of species A (C_{A0}) are chosen to be the manipulated inputs.

The states and manipulated inputs are converted into deviation variable form, meaning we denote the state variables as $x = [C_A - C_{As},\ T - T_s]$ and the input variables as $u = [C_{A0} - C_{A0s},\ \dot{Q} - \dot{Q}_s]$. These deviation variables are defined with respect to the unstable steady-state values of the respective terms. For the input variables, we introduce upper and lower bounds of $[-3.5,\ 3.5]$ $kmol/m^3$ and $[-5 \times 10^5,\ 5 \times 10^5]$ kJ/hr for the two elements of u respectively. All simulations begin at the unstable steady-state at $t_0 = 0$ and optimize a cost function that varies with time based on the consumption of species A, production of species B, and the current deviation form heat input rate $\dot{Q} - \dot{Q}_s$ as shown in Eq. (13)

$$L(x_E, u) = A_1 k_0 e^{-\frac{E}{RT}} C_A^2 - A_2(C_{A0} - C_{A0_s}) - A_3(\dot{Q} - \dot{Q}_s)^2 \tag{13}$$

To simulate the evolving economics of operation, this cost function includes A terms which are time-varying weights for each component as defined in Table 2. The first-principles model shown in Eq. (12) are used in both the MPC as the process model shown in Eq. (11b) and in the trajectory estimator that is used to generate the set-points for the lower-layer controller.

4.2. Stability analysis

The control Lyapunov function for this system is of the form $V(x) = x^\top P x$ where P is a positive definite matrix defined as:

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{14}$$

Table 2
Time-varying LEMPC weights for chemical process example.

Time (t)	A_1	A_2	A_3
$0\,\text{hr} \le t < 1\,\text{hr}$	1	17	1×10^{-8}
$1\,\text{hr} \le t < 2\,\text{hr}$	0.99	14	0.8×10^{-8}
$2\,\text{hr} \le t < 3\,\text{hr}$	1.01	5	0.84×10^{-8}
$3\,\text{hr} \le t < 4\,\text{hr}$	0.98	7	0.9×10^{-8}
$t \ge 4\,\text{hr}$	1.02	9	0.9×10^{-8}

Stability can be guaranteed within a level set of this Lyapunov function, denoted Ω_ρ with a value of $\rho = 135$. This level set was determined by simulating the system with the reference stabilizing controller. Starting at a large V, 100 initial conditions were sampled along the level set boundary. For each initial condition, the closed-loop system under the reference controller was simulated for 10 sampling times, and $\dot{V}$ was checked at each. Once $\dot{V}$ returned negative values for all 10 time steps for all 100 points, the boundary of the stability region of the stabilizing controller was deemed found. To allow for sufficient room for disturbances and cyberattack detection, a deeper sub-region $\Omega_{\rho_{secure}}$ is chosen where $\rho_{secure} = 85$.

The reference stabilizing controller used to determine the stability region is a set of PI controllers for C_{A0} and $\dot{Q}$ with proportional gains of $K_{C_{A0}} = 10^4$ and $K_Q = 10$ and integral time constants of $\tau_{C_{A0}} = 10^{-6}$ and $\tau_{\dot{Q}} = 10^{-3}$, respectively. These controllers used a sampling time of $\Delta_{PI,\text{reference}} = 7.2\,\text{s}$. The lower-layer controllers used in this example share the same gains, but operate with $\Delta_{PI} = 0.36\,\text{s}$.

Remark 3. Even if the lower-layer controller was used as the reference, we found that the same stability region is valid. The stability region is a conservative region within the true stability region of the closed-loop system. Further, for ease of computation, a P controller with proportional gains $K_{C_{A0}} = 2$ and $K_Q = 12{,}000$ was used as the reference controller for the LEMPC constraint shown in Eq. (11f).

4.3. Control system parameters

In order to apply encryption to a process, the sampling time of the controller must sufficiently exceed the computation time associated with the solution of the control inputs as well as the encryption–decryption process. In other words, Δ_{MPC} must exceed the time it takes to solve the LEMPC once, solve and sample the estimated state trajectories, encrypt, and then transmit the relevant set-points to the lower-layer. Δ_{PI} must then exceed the computation and decryption time. The delay involved with signal transmission in a networked environment is neglected, as magnitude of this delay is insignificant relative to the other parameters. These requirements must be fulfilled for both the upper and lower-layer control systems in an encrypted two-layer control architecture and can be expressed as follows:

$$\Delta_i > \max(\text{Enc time})_i + \max(\text{Dec time})_i + \max(\text{Computation time})_i$$
$$\text{Tier index}: \; i = \{1, 2\} \tag{15a}$$

Here Δ_1 and Δ_2 represent the lower(Δ_{PI}) and upper(Δ_{MPC}) control layer, respectively. The closed-loop system will be simulated over a period of 5 hrs, during which the objective function weights change in intervals of 1 hr. The LEMPC continuously optimizes for a horizon length of 432 s with a sampling period of $\Delta_{MPC} = 7.2\,\text{s}$ which corresponds to a horizon length of $N = 60$. The sampling period was selected to be sufficiently small to yield smooth trajectories and minimize the risk of instability induced by the sample-and-hold implementation of the control signals. Additionally, the sampling period was selected to be sufficiently large to allow ample room for computation as is required as discussed above. This is enabled by the low-dimensionality of the process and the simplicity of the encryption key – as mentioned in Remark 4 – which results in low net computation times. The optimal control inputs are then applied to a simulated process model of Eq. (12) to yield the predicted state-trajectory. A total of 5 set-points are sampled from this trajectory in constant intervals of 1.44 s. Sensor readings for the state variables are passed to the PI controller in intervals of 0.36 s.

To achieve sufficient precision while maintaining low computational overhead, the LEMPC uses the forward explicit Euler method with an integration step size ($h_{c,LEMPC}$) of 0.36 s. The PI controller follows the same numerical simulation structure, but with an integration step size ($h_{c,PI}$) of 0.036 s.

4.4. Encryption

To encrypt the sensor–controller–actuator signals, we need to select a sufficiently large d, and appropriate l and M values. The selection of d depends on the desired precision of data, which we chose to be $d = 12$. l needs to be chosen such that $\mathbb{Q}_{l,d}$ encompasses the range of expected data values that will be encrypted. Large terms such as the heat input would dictate the smallest value of l, but since we use PythonPaillier (Data61, 2013) in our design, the selection of l is done automatically based on M.

Remark 4. To ensure robustness, NIST recommends using keys with at least 2048 bits for authorization data, and 256 bits for less important data, such as conversation keys (Ferraiolo and Regenscheid, 2024). Increasing the key length increases computation time, so it is important to secure the signals to as high of a key length as the hardware is capable of. For reference, we used PythonPaillier's default, where M is some 256-bit long integer.

4.5. Closed-loop simulation results

We use the vector γ to denote the absolute value of the ratio of the difference between two consecutive set-point values sent to the lower-layer over the time difference between two consecutive set-point changes. The closed-loop system results where $\gamma = [+\infty + \infty]$ are seen in Figs. 2 and 5 where the progression through time on the state-space plot is depicted by a color change from green to red. From this plot, we can see the impact of the time varying weights on the closed-loop system response and economic performance. When γ becomes finite such that the difference between two consecutive set-point values over the time difference between two consecutive set-point changes is limited to 100 for C_A and 10,000 for T (i.e., $\gamma = [100\ \ 10,000]$), we obtain the results in Figs. 3 and 6. As this set-point rate of change constraint is further tightened, the maximum set-point change drops to 50 for C_A and 5000 for T (i.e., $\gamma = [50\ 5,000]$), which yields the results shown in Figs. 4 and 7. Notably, we do not apply constraints to the rate of change of the control actions directly in this example.

It can be seen in all results that the time-varying economics results in roughly three operating zones where optimal operation is found. The influence of tightening the set-point rate of change constraint does not change the existence of these zones, but instead indirectly limits how sharp the state-trajectory is allowed to change by limiting the rate of change for the controllers' set-points. Due to the PI control's imperfect tracking of the optimal state-trajectory, the tightening of the set-point rate of change constraint results in marginally improved tracking up to a certain point, which yields small improvements to the net objective function as seen in Table 3.

Although there seems to be a slight trend towards better cost with tightening of the set-point rate of change constraint, this trend falls apart as the tightening increases, as expected. Because we only implement the first control input of the LEMPC, after every 7.2 s the set-points are reset with respect to the current set-point values. This sudden shift leads to jagged movement in the set-points on the initial point of each LEMPC control input. Because of this, the process dynamics

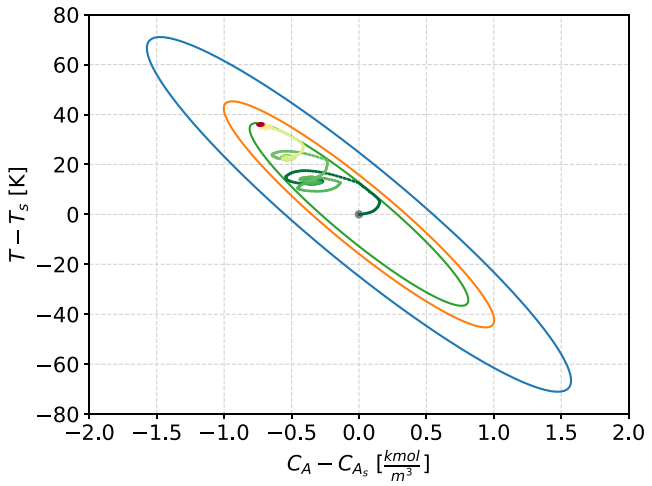

Fig. 2. State-space plot of the closed-loop trajectory for $\gamma = [\infty, \infty]$.

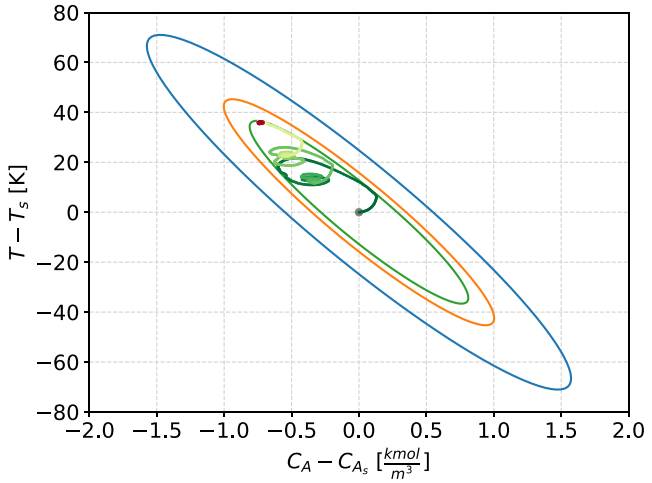

Fig. 3. State-space plot of the closed-loop trajectory for $\gamma = [100\ 10,000]$.

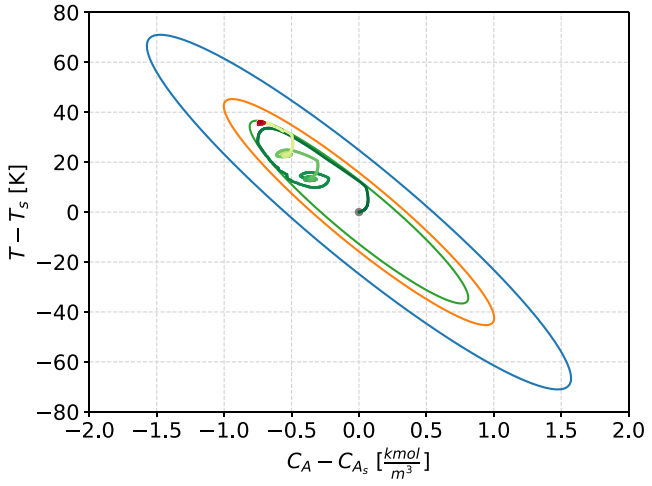

Fig. 4. State-space plot of the closed-loop trajectory for $\gamma = [50\ 5,000]$.

and poor tracking of the PI controllers can lead to gradual divergence from the economic and stability regions. These phenomena are shown in Figs. 8 and 9 which are simulations where the maximum set-point rate of change is 1 for C_A and 100 for T (i.e., $\gamma = [1\ 100]$).

Table 3
Impact of γ on economic objective values.

Max set-point rate of change		Net objective function	Increase (%)
C_A	T		
$\pm\infty$	$\pm\infty$	65.165	27.39
±100	$\pm10,000$	66.361	29.73
±50	$\pm5,000$	65.944	28.92
±1	±100	59.786	16.88
0	0	51.153	00.00

Remark 5. By design, the two-layer implementation will yield varying results depending on the ability of the PI controllers regarding tracking the optimal state-trajectories. Due to the differences between the actual and constrained set-point trajectories, the observed divergence due to significant tightening of the set-point change is unavoidable at extreme values, and thus it is suggested that significant tightening is used sparingly.

5. Application to a chemical process modeled by an Aspen Plus simulator

This section showcases the practical application of the proposed encrypted two-layer control framework in the context of a large-scale chemical process modeled using an Aspen simulator. A first-principles based model is initially built to represent this large-scale process in a manner that can be used as the prediction model within an LMPC (in this study a quadratic, tracking cost function is employed). Using this first-principles model, the optimal control from the LMPC is used to predict the future trajectory of the state variables for the process. In the lower-layer, a fully encrypted PI controller is used, and the trajectory solved in the upper-layer (where the LMPC is used) is sampled to provide set-points that the lower-layer PI controllers will use. The resulting control action is applied to a second dynamic model of the same process built using Aspen Plus Dynamics. Unlike traditional steady-state solvers, Aspen Plus Dynamics is a dynamic process simulator that allows for the detailed simulation of the time-varying dynamics of complex processes. This allows for the control actions to be applied to a process that mimics reality as close as possible without the need to construct a physical experimental process. The upper-layer LMPC computations can be done in a SCADA system using Python with readily available numerical integration and optimization solvers. By applying the control actions of the PI controllers to the Aspen model, but solving the upper-layer using a first-principles based LMPC and using a lower-layer to calculate the control actions implemented on the process, we can simultaneously explore the dynamics arising from modeling, tracking, and quantization error. To demonstrate the improved performance relative to linear control, and the improved cyber resilience with minimal performance loss relative to nonlinear control clearly, the closed-loop system performance under the two-layer architecture is compared to an unencrypted tracking PI controller and tracking MPC system.

5.1. 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. (16) and takes place in two non-isothermal, well-mixed continuous stirred tank reactors (CSTR). The chemical reactions involved are as follows:

$$C_2H_4 + C_6H_6 \longrightarrow C_8H_{10} \quad \text{(primary)} \tag{16a}$$

$$C_2H_4 + C_8H_{10} \longrightarrow C_{10}H_{14} \tag{16b}$$

$$C_6H_6 + C_{10}H_{14} \longrightarrow 2\,C_8H_{10} \tag{16c}$$

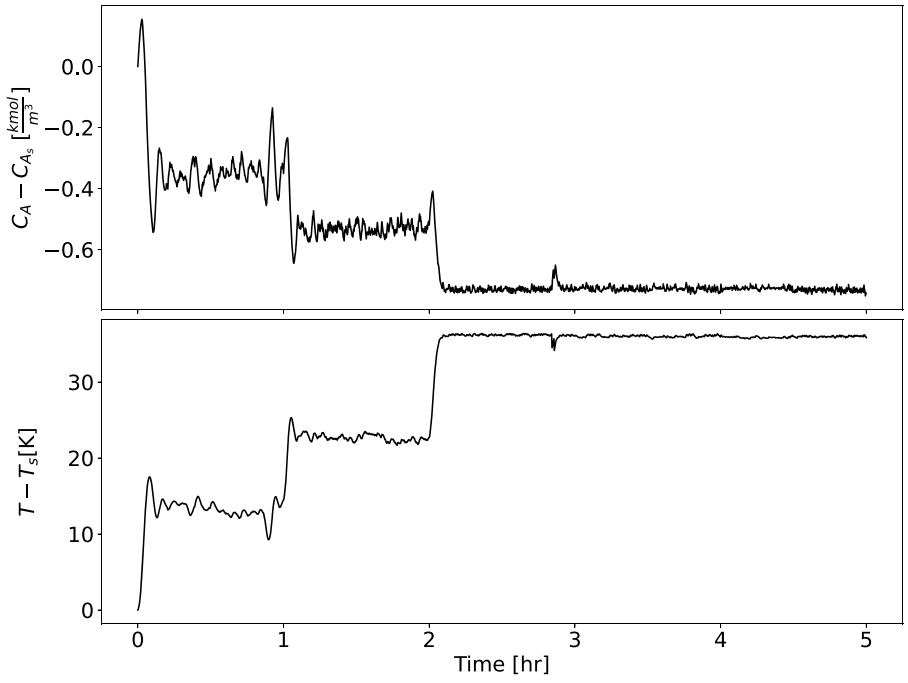

Fig. 5. Closed-loop states for $\gamma = [\infty\ \infty]$.

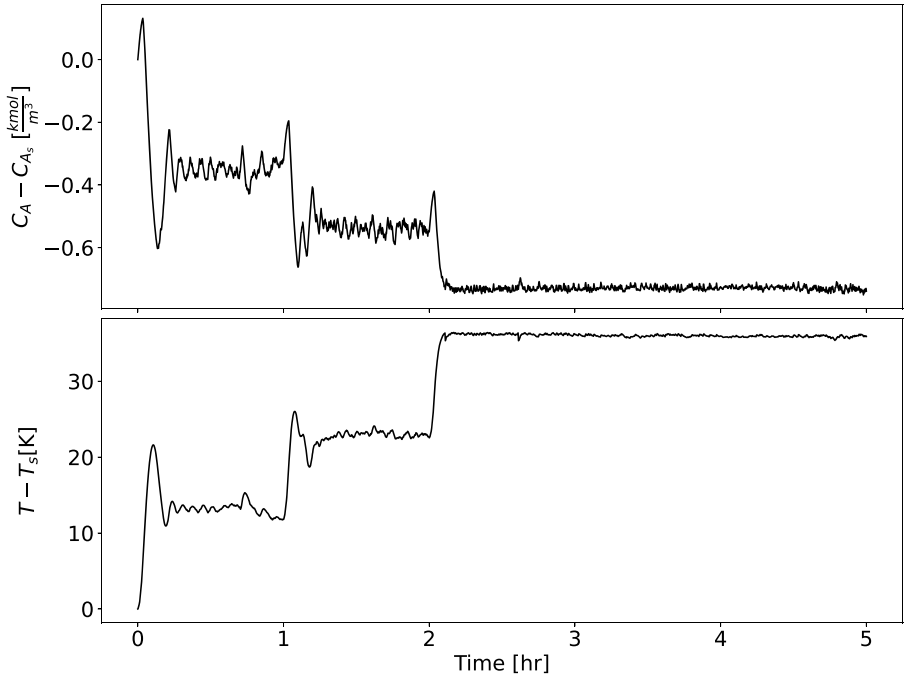

Fig. 6. Closed-loop states for $\gamma = [100\ 10,000]$.

The state variables are the concentration of Ethylene, Benzene, Ethylbenzene, Di-Ethylbenzene and the reactor temperature, for each 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 rates of heat removal for each reactor $[Q_1 - Q_{1s},\ Q_2 - Q_{2s}]$ are the control inputs manipulated by the lower-layer using an encrypted PI control system (two PIs), which are bounded by the closed sets $[-4 \times 10^3$kW, $5(\times 10^3 - Q_{1s})$kW] and, $[-3 \times 10^4$kW, $(5 \times 10^3 - Q_{2s})$kW] respectively. The bounds of the heat input terms were designed in a way that is feasible to implement in practice. The lower bounds are roughly 10× what is necessary to maintain the first-principles model at the desired unstable steady-state, and the upper bounds are roughly the amount of heat necessary to heat up 60 m^3 of pure water by 80 K in 1 hr.

The control objective is to move from a stable steady-state with poor economic performance to an unstable steady-state with higher economic performance in a stable and efficient manner. This is required for both CSTRs simultaneously, and both reactors must remain at this unstable steady-state. To identify the stability condition of the starting steady-state and the operating steady-state, we conducted two open-loop simulations in Aspen Plus Dynamics for a total simulation of

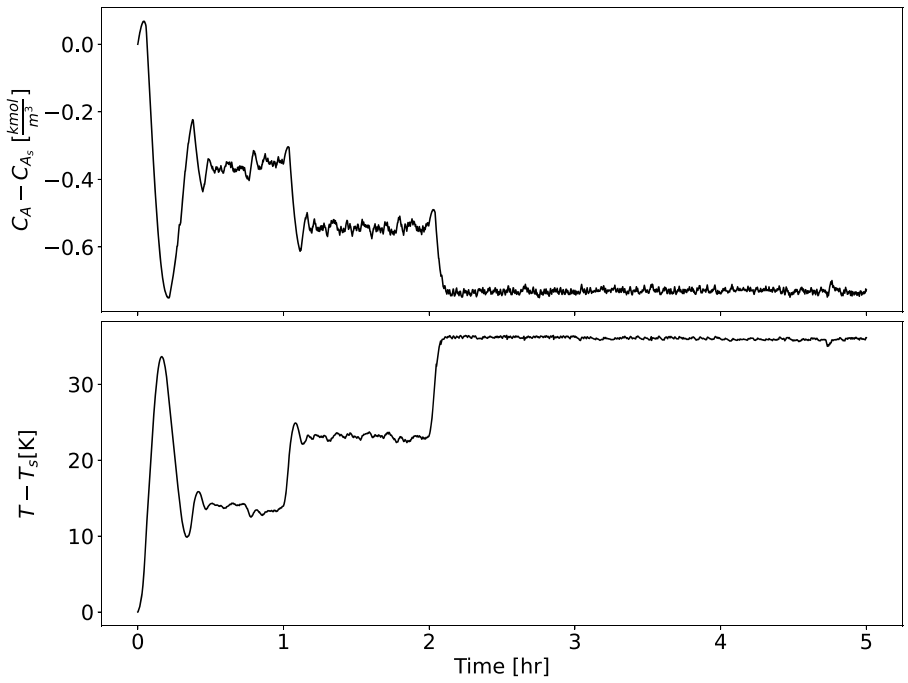

Fig. 7. Closed-loop states for $\gamma = [50\ 5,000]$.

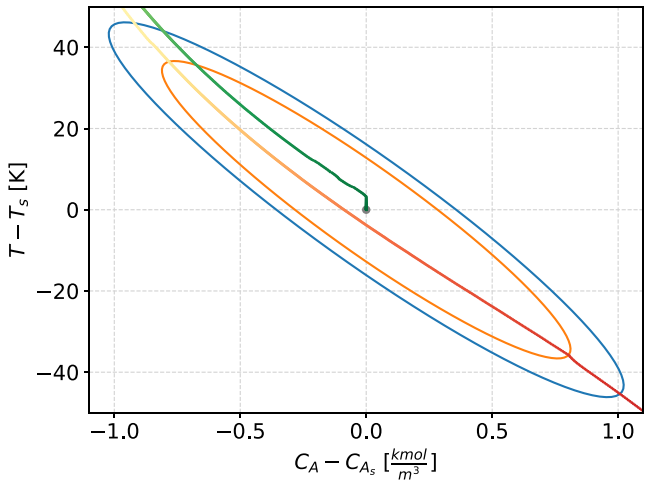

Fig. 8. State-space plot of the closed-loop trajectory for $\gamma = [1\ 100]$ without stabilizing control.

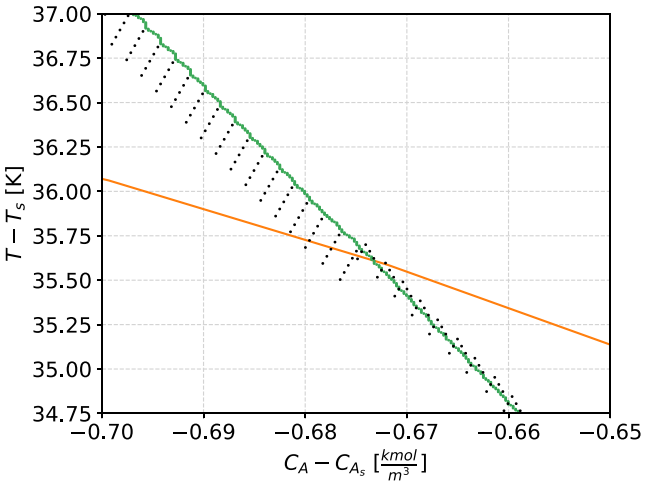

Fig. 9. Optimal versus constrained set-point values used by the PI controllers. Orange is the secure region boundary, Green is the actual state-trajectory, and black is the set-points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

process time of 6 h. In both cases, the state variables and manipulated inputs were initialized to the stable steady-state values. At this point, the system was run using the Dynamic option within Aspen Plus Dynamics with the manipulated inputs remaining fixed for the full duration. The starting steady-state remained approximately the same after the simulation, whereas the operating steady-state stabilized to a distinct steady-state, providing clear evidence that the selected operating condition is an unstable steady-state whereas the starting steady-state is a stable steady-state. The main reason behind choosing this operating state was due to the increased yield of Ethylbenzene, our desired product.

5.2. Aspen Plus dynamics model development

In order to model a process to a degree of accuracy that closely resembles a real-world implementation of the two-layer control system, it is necessary to use high-fidelity simulators such as Aspen Plus Dynamics V12. Our process is first modeled inside Aspen Plus to determine the starting and operating steady-states or the process before being converted into an Aspen Plus Dynamics model that functions using pressure driven flow. The construction of the Aspen Plus model is as follows:

1. Properties:
   (a) Low temperature and high pressures ensure that the only valid phase is Liquid.
   (b) Component Specification (CAS numbers):
      i. 71-43-2
      ii. 141-93-5
      iii. 74-85-1

Table 4
Parameter values, steady-state values, and model configuration of the Aspen model.

$T_{1o} = T_{2o} = 300\,\text{K}$	$T_{1d} = 350\,\text{K}$	$T_{1s} = 300\,\text{K}$
$V_1 = V_2 = 60\,\text{m}^3$	$T_{2d} = 400\,\text{K}$	$T_{2s} = 300\,\text{K}$
$C_{B_{o1}} = 2\,\text{kmol/m}^3$	$C_{B_{1d}} = 5.73\,\text{kmol/m}^3$	$C_{B_{1s}} = 6.96\,\text{kmol/m}^3$
$C_{E_{o1}} = 7\,\text{kmol/m}^3$	$C_{DEB_{1d}} = 3.93 \times 10^{-5}\,\text{kmol/m}^3$	$C_{DEB_{1s}} = 3.10 \times 10^{-8}\,\text{kmol/m}^3$
$C_{B_{o2}} = 2\,\text{kmol/m}^3$	$C_{E_{1d}} = 0.954\,\text{kmol/m}^3$	$C_{E_{1s}} = 1.96\,\text{kmol/m}^3$
$C_{E_{o2}} = 6\,\text{kmol/m}^3$	$C_{EB_{1d}} = 0.956\,\text{kmol/m}^3$	$C_{EB_{1s}} = 0.042\,\text{kmol/m}^3$
$E_1 = 71.160\,\text{kJ/kmol}$	$C_{B_{2d}} = 4.23\,\text{kmol/m}^3$	$C_{B_{2s}} = 6.44\,\text{kmol/m}^3$
$E_2 = 83.680\,\text{kJ/kmol}$	$C_{DEB_{2d}} = 2.16 \times 10^{-4}\,\text{kmol/m}^3$	$C_{DEB_{2s}} = 2.82 \times 10^{-8}\,\text{kmol/m}^3$
$E_3 = 62.760\,\text{kJ/kmol}$	$C_{E_{2d}} = 0.171\,\text{kmol/m}^3$	$C_{E_{2s}} = 1.96\,\text{kmol/m}^3$
$k_1 = 1.528 \times 10^6\,\text{m}^3\,\text{kmol}^{-1}\,\text{s}^{-1}$	$C_{EB_{2d}} = 1.64\,\text{kmol/m}^3$	$C_{EB_{2s}} = 0.039\,\text{kmol/m}^3$
$k_2 = 2.778 \times 10^4\,\text{m}^3\,\text{kmol}^{-1}\,\text{s}^{-1}$	$Q_{1d} = -412.254\,\text{kW}$	$Q_{1s} = -56.4623\,\text{kW}$
$k_3 = 0.4167\,\text{m}^3\,\text{kmol}^{-1}\,\text{s}^{-1}$	$Q_{2d} = -733.54\,\text{kW}$	$Q_{2s} = -56.2121\,\text{kW}$
$F_1 = 43.2\,\text{m}^3/\text{h}$	$F_2 = 47.87\,\text{m}^3/\text{h}$	$R = 8.314\,\text{kJ}\,\text{kmol}^{-1}\,\text{K}^{-1}$
$\rho_1 = 639.153\,\text{kg/m}^3$	$\rho_2 = 607.504\,\text{kg/m}^3$	$C_p = 2.411\,\text{kJ}\,\text{kg}^{-1}\,\text{K}^{-1}$
$\Delta H_1 = -1.04 \times 10^5\,\text{kJ/kmol}$	$\Delta H_2 = -1.02 \times 10^5\,\text{kJ/kmol}$	$\Delta H_3 = -5.5 \times 10^2\,\text{kJ/kmol}$

'o' subscript stands for inlet flow values
'd' subscript stands for 'desired', i.e. the point being tracked
's' subscript stand for the initial steady-state values

iv. 100-41-4
v. 110-54-3

(c) The PSRK method is chosen. The interested reader may refer to Table 4 for stream properties. The purpose of Hexane in this system is to serve as a solvent.

2. Simulation:

(a) Construct the Main Flowsheet to match Fig. 10.
(b) Construct the Reactions to match the forms from Eq. (16) with parameter values specified in Table 4. The implementation is as irreversible elementary power-law reactions.
(c) CSTRs,Valves, and Feed streams are designed to match the specifications shown in Table 4. The valves and pumps are key elements, as the process' flow is pressure driven, and these valves serve as pressure regulators. Valves 1,2, 3 and 4 drop the pressure by 5, 5, 2, and 14 bars respectively. The Pump sets the discharge pressure to 15 bars. The feed streams all operate at a pressure of 20 bars.

3. Once fully prepared, Aspen Plus is used to solve for the steady-state that defines the initial process state.
4. A pressure checker tool is available in Aspen Plus to ensure no pressure issues are present before converting to an Aspen Pus Dynamics file via the Pressure Driven option after running once with Dynamic mode steady-state.
5. The resulting Dynamics file will automatically add controllers for both the fluid level and temperatures in both reactors. Since we control the heating rates, the temperature controllers are removed in favor of constant heating duty. By running the simulation for fixed time intervals using scripts, followed by modifying the constant heating duty, a sample-and-hold implementation of control is applied.

5.3. First-principles model development

The first-principles model for the CSTRs is formulated by employing the theory of mass and energy balances. The dynamic model of the first CSTR is described by the following ODEs:

$$\frac{dC_{E_1}}{dt} = \frac{(F_1 C_{Eo_1} - F_{out_1} C_{E_1})}{V_1} - r_{1,1} - r_{1,2} \tag{17a}$$

$$\frac{dC_{B_1}}{dt} = \frac{(F_1 C_{Bo_1} - F_{out_1} C_{B_1})}{V_1} - r_{1,1} - r_{1,3} \tag{17b}$$

$$\frac{dC_{EB_1}}{dt} = \frac{-F_{out_1} C_{EB_1}}{V_1} + r_{1,1} - r_{1,2} + 2r_{1,3} \tag{17c}$$

$$\frac{dC_{DEB_1}}{dt} = \frac{-F_{out_1} C_{DEB_1}}{V_1} + r_{1,2} - r_{1,3} \tag{17d}$$

$$\frac{dT_1}{dt} = \frac{(T_{1_o} F_1 - T_1 F_{out_1})}{V_1} + \sum_{i=1}^{3} \frac{-\Delta H_i}{\rho_1 C_p} r_{1,i} + \frac{Q_1}{\rho_1 C_p V_1} \tag{17e}$$

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

$$\frac{dC_{E_2}}{dt} = \frac{(F_2 C_{Eo_2} + F_{out_1} C_{E_1} - F_{out_2} C_{E_2})}{V_2} - r_{2,1} - r_{2,2} \tag{18a}$$

$$\frac{dC_{B_2}}{dt} = \frac{(F_2 C_{Bo_2} + F_{out_1} C_{B_1} - F_{out_2} C_{B_2})}{V_2} - r_{2,1} - r_{2,3} \tag{18b}$$

$$\frac{dC_{EB_2}}{dt} = \frac{F_{out_1} C_{EB_1} - F_{out_2} C_{EB_2}}{V_2} + r_{2,1} - r_{2,2} + 2r_{2,3} \tag{18c}$$

$$\frac{dC_{DEB_2}}{dt} = \frac{F_{out_1} C_{DEB_1} - F_{out_2} C_{DEB_2}}{V_2} + r_{2,2} - r_{2,3} \tag{18d}$$

$$\frac{dT_2}{dt} = \frac{(T_{2_o} F_2 - T_1 F_{out_1} - T_2 F_{out_2})}{V_2} + \sum_{i=1}^{3} \frac{-\Delta H_i}{\rho_2 C_p} r_{2,i} + \frac{Q_2}{\rho_2 C_p V_2} \tag{18e}$$

where the reaction rates are determined using the following expressions:

$$r_{n,1} = k_1 e^{\frac{-E_1}{RT_n}} C_{E_n} C_{B_n} \tag{19a}$$

$$r_{n,2} = k_2 e^{\frac{-E_2}{RT_n}} C_{E_n} C_{EB_n} \quad n = 1, 2 \text{ (reactor index)} \tag{19b}$$

$$r_{n,3} = k_3 e^{\frac{-E_3}{RT_n}} C_{DEB_n} C_{B_n} \tag{19c}$$

5.4. Implementing encryption in the two-layer control architecture

Before implementing encryption–decryption in the closed-loop system, it is crucial to carefully choose the values: d, and l. l is determined by looking at the maximum and minimum permissible values of the states and inputs, and from these data the number of integer bits, $l-d$, is determined. Recall from Eq. (4) that the largest value in the set $\mathbb{Q}_{l,d}$ is obtained using the formula $2^{l-d-1} - 2^{-d}$, while the smallest value is -2^{l-d-1}. From this, it can be seen that l must be sufficiently large, and that d, the quantization parameter, should be selected based on factors such as desired accuracy and the range of state and input values. Additionally, the bit length for the encryption keys must be selected to be larger than l. Considering that NIST recommends a key size of 2048 for authorization data or 256 for less important data, we can simply set l equal to the key's bit length to ensure sufficient room for arithmetic operation (Ferraiolo and Regenscheid, 2024). For the sake of limiting computational complexity, $l = 256$ is used, although any sufficiently large value can be used so long as the resulting delay from computation

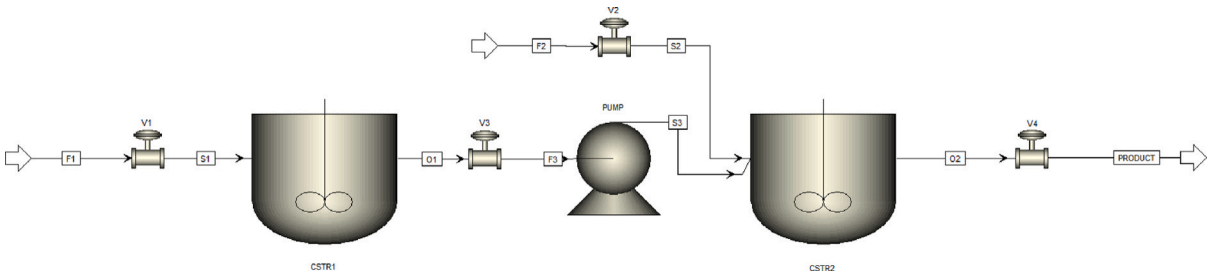

Fig. 10. Aspen Plus Dynamics model flow sheet.

time is within insignificant margins. In the set $\mathbb{Q}_{l,d}$, 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 use the quantization parameter $d = 12$ as performing encryption using a higher quantization parameter provides nearly identical trajectories as the case without encryption (Suryavanshi et al., 2023). After determining all the quantization parameters, the implementation of Paillier Encryption is carried out using the "phe" module in Python, specifically Python-Paillier (Data61, 2013). This module includes encryption, decryption, and encoding processes which automatically apply the quantization and mapping described above.

5.5. Implementing the fully-encrypted lower-layer controller

In the lower-layer, since linear control is used, control input computations are confined to linear addition and multiplication operations, ensuring their execution within a fully-encrypted domain that guarantees security against cyberattacks during networked communication. The selection of lower-layer controlled inputs, which possess the capability to stabilize the entire system, is a pivotal task that requires adherence to a well-defined procedure. For the purposes of this example, the critical criteria to meet was a minimization of overshoot in the Temperature to ensure the risk of thermal runaway is minimized given a fixed sampling time for the linear controllers of $\Delta_{PI} = 3\,\text{s}$ along with demonstrated tracking capabilities with no-overshoot. To achieve this, rigorous simulations of the nonlinear chemical process model were done until the criteria from above was met. For the given process, a natural phenomena occurs at high temperature wherein the reactions run to completion, which results in a sharp Temperature drop due to the lack of heat generated from the exothermic reactions. Controller gains were increased until the results demonstrated the ability to limit the overshoot below the temperature point in which this phenomena occurs. This was done for both the stabilizing P controller used in the LMPC for use in equation Eq. (11f) as well as the PI controllers that are used for comparisons as seen in Section 5.8. For the PI controller that is used in the two-layer design, the same criteria were used, but additional fine tuning was done through simulations using the Aspen Plus Dynamics model until the controller demonstrated sufficient tracking capabilities.

5.6. Implementation of the upper-layer LMPC

The first-principles model, expressed by equations in Eqs. (17) and (18), serves as the foundational process model within the LMPC framework. These first-principles model is used in both the MPC as the process model shown in Eq. (11b) and in the trajectory estimator that is used to generate the set-points for the lower-layer controller. For solving the optimization problem, we leverage the SciPy Python module's minimize function (Virtanen et al., 2020). More specifically, we use the 'SLSQP' method to solve the LMPC optimization problem at each sampling time. The optimization problem at hand falls under the category of constrained non-convex optimization problems. Consequently, the resultant solution is a local optimum rather than a global one. The control inputs chosen for manipulation by the lower-layer controller consist of the heating rate for each CSTR. This results in a total of 2 control variables that are to be optimized over a prediction horizon of length 10.

The process of resolving this optimization problem involves defining constraints for the LMPC. The optimization problem operates within a feasible region and employs an iterative methodology to progressively navigate towards the optimal solution by traversing the interior of the feasible region. The LMPC achieves this while utilizing two distinct operating modes. Specifically, while within the feasible region, every future state within the prediction horizon is constrained to also be within the feasible region. This operating mode is described in Eq. (11e). While outside the feasible region, but within the stability region, the initial slope of the Lyapunov function is constrained to be more-negative than some reference stabilizing controller. This operating mode is described in Eq. (11f). These parameters function as the primary constraints of the optimization problem. If the optimization fails for any reason, the system utilizes the control input calculated by the backup controller as the solution.

5.7. Sampling time criteria

To implement encryption in a practical setting, it is crucial to ensure that the upper-layer's sampling time, Δ_{MPC} exceeds the combined computation time of the LMPC solution as well as the future trajectory simulation required for the lower-layer control system set-points. Additionally the lower-layer's sampling time Δ_{PI} must exceed the combined maximum of the encryption–decryption time as well as the maximum time needed for computing the control action. The delay involved with signal transmission in a networked environment is neglected, as magnitude of this delay is insignificant relative to the other parameters. These requirements must be fulfilled for both the upper and lower-layer control systems in an encrypted two-layer control architecture and can be expressed as follows:

$$\Delta_i > \max\left(\text{Enc time}\right)_i + \max\left(\text{Dec time}\right)_i + \max\left(\text{Computation time}\right)_i$$
$$\text{Tier index}: \; i = \{1, 2\} \tag{20a}$$

Here Δ_1 and Δ_2 represent the lower(Δ_{PI}) and upper(Δ_{MPC}) control layer, respectively. Notably, since the upper-layer is not encrypted, but the set-points that are transmitted are, the decryption time is 0. The sampling time, Δ_{MPC} is carefully selected as 30 s, considering the aforementioned condition to ensure proper implementation. 30 s was deemed sufficiently long to allow for both MPC computation as well as encryption based off of observed computation times during the design process of the MPC. Existing literature that uses the same chemical process example has demonstrated similar results, with encrypted centralized MPC computations taking 15.28 s on average (Kadakia et al., 2023). Experimental validation demonstrated sufficiently smooth trajectories for the process without the presence of excessive oscillatory behavior, thus the sampling time is deemed sufficient for the purposes of demonstration.

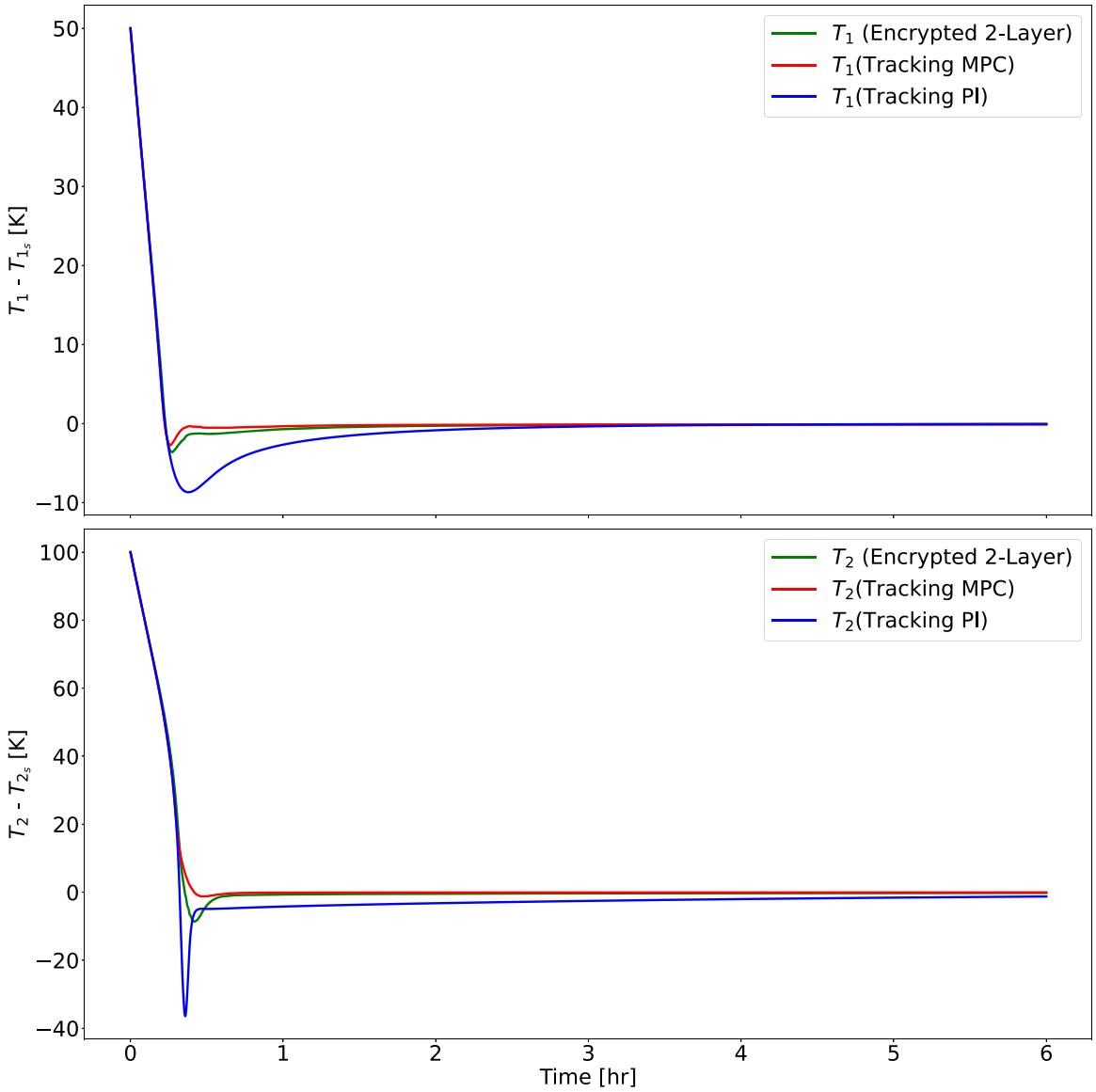

Fig. 11. Closed-loop temperature profiles under the three different control strategies.

Based on these results, the integration step h_c is chosen as $(10^{-2} \times \Delta)$ to evaluate the cost function of the LMPC through the first-principles model. Because the lower-layer operates over a smaller step size of $\Delta_{PI} = 3\,\text{s}$, we use a proportionally smaller integration step size of $h_{c_{PI}} = 10^{-2} \times \Delta_{PI}$. Additionally, the set-points are sampled from the future trajectory of length $10\Delta_{MPC}$ in increments of $\Delta_{Set-Point} = 120\,\text{s}$. Because we only apply the first MPC control action before recalculating the trajectory, only the first set-point is utilized, and these set-points are updated in intervals of Δ_{MPC}. The positive definite matrix P in the control Lyapunov function $V = x^T Px$ for this system is taken as $P = \text{diag}[200,\ 250,\ 2500,\ 10,\ 0.25,\ 1000,\ 1000,\ 500,\ 1,\ 0.5]$ based on extensive simulations. A prediction horizon of $N = 10$ is employed in the LMPC framework. To ensure stability in the LMPC, we set the criterion $\rho_{min} = 450\,\text{K}$. Additionally, a contractive constraint of the form $\dot{V}_{MPC} \leq \dot{V}_P$ is utilized for Eq. (11f), where $\dot{V}_P$ is the stabilizing backup Proportional controller with gains $[500, 500]$ that was designed as described in Section 5.6. 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 $Q_2 = \text{diag}[5 \times 10^{-6},\ 1.25 \times 10^{-5}]$, respectively. The cost function is defined as $L(x(t), u(t)) = x^T Q_1 x + u^T Q_2 u$.

5.8. Closed-loop simulation results

The Aspen Plus Dynamics model is simulated with 3 forms of control: tracking PI control to drive the process to temperatures 350 K and 400 K in reactors 1 and 2, respectively, tracking MPC to the same steady-state, and the two-layer design for tracking to the same steady-state as well. The results, as seen in Figs. 11 to 13, demonstrate that tracking MPC has minimal temperature overshoot and jaggedness in the control actions as expected from the chosen cost-function. The tracking PI demonstrates smoother control action at the expense of significant overshoot in the Temperature, which puts the reactor at greater risk of thermal runaway. The two-layer architecture yields results that are very similar to the tracking MPC design despite having encrypted control. Thus, the goal of enhanced cybersecurity with minimal impact on process performance has been achieved for the highly nonlinear Aspen Dynamics model.

The most notable consequence of the proposed two-layer design is the visible jaggedness of the control action due to the shifting set-points as the reactor heats up. As was true in Section 4, the nature of the shifting set-points introduces problems with the PI controllers tracking. Fig. 9 demonstrates the divergent behavior of poor tracking

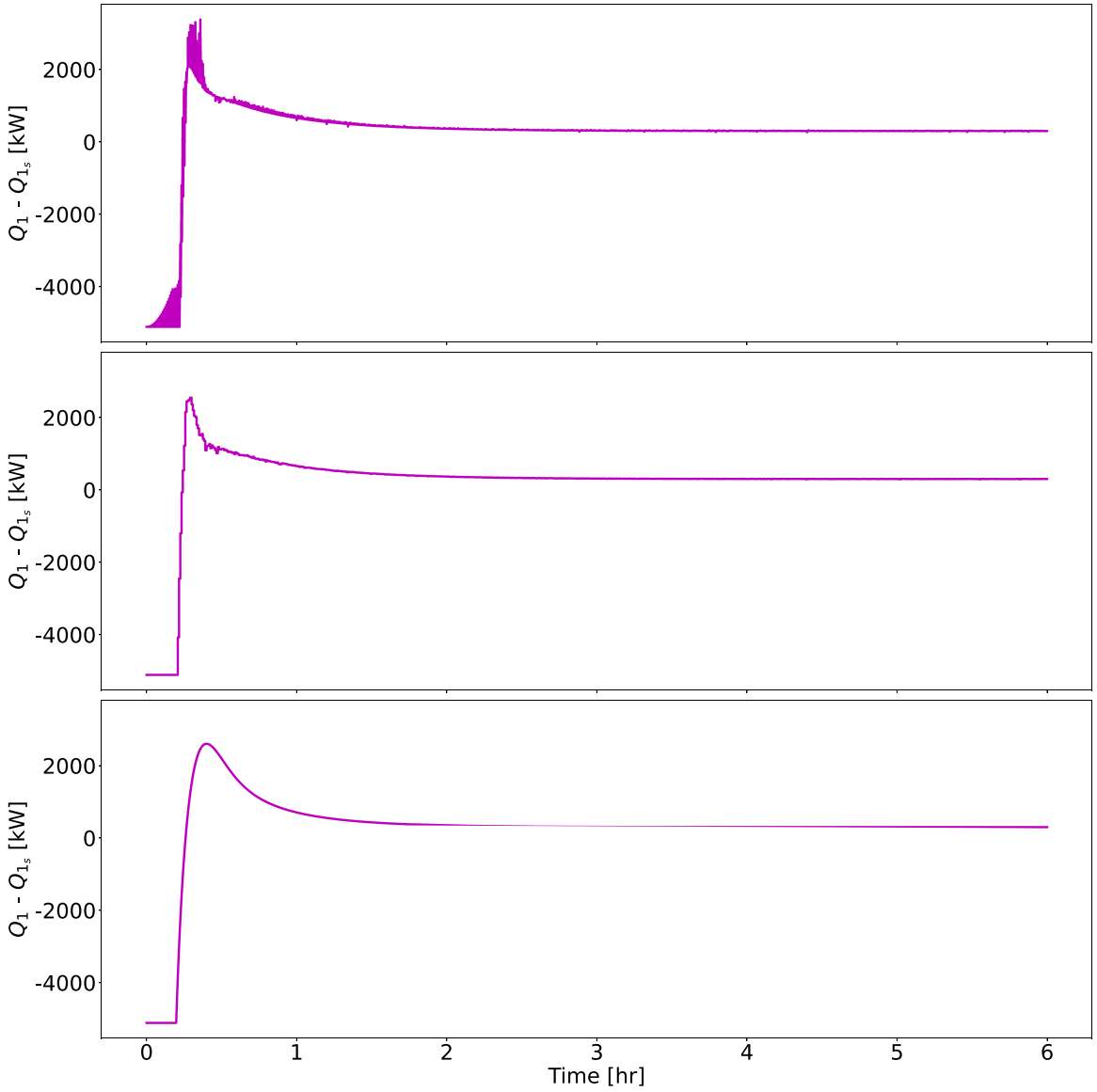

Fig. 12. Manipulated input profiles for the first CSTR control action (the top plot is the input under two-layer control, the middle plot is the input under tracking MPC and the bottom plot is the input under PI control only).

in the two-layer design, but at a closer glance, it can be seen that it also demonstrates a sharp transition in the set-point trajectory after each recalculation of the optimal control inputs in the upper-layer. When the MPC is resolved, the set-point that the lower-layer is tracking will suddenly shift to a new value. In the early stages of the systems' response, the temperature variable experiences significant changes as the reactors heat up. This large jump in the set-point, when the PI control is capable of tracking, results in sudden bursts of control action that gradually weaken only to spike again when the set-point moves once more. This problem was mitigated as described in Section 3.3 by bounding the change in magnitude of the control inputs based on the magnitude of the state variable, but it could be further mitigated as described in Section 3.4 by bounding the change in the set-points themselves. Both constraints mitigate the issue of jaggedness but come at the expense of longer settling time and a higher risk of over-constraining the control action to the point of risking closed-loop instability.

6. Conclusion

This work proposed a two-layer multi-key control framework that is implemented with nonlinear Lyapunov-based (economic) model predictive controller (LEMPC) in the upper-layer and linear proportional integral (PI) control in the lower-layer to allow for cyber-secure operation while allowing for economic optimization. The upper-layer LEMPC solves optimal control inputs, which are used to simulate the optimal closed-loop state-trajectory. This trajectory is sampled and used as set-points for the lower-layer PI control system to use in order to improve economic operation. Due to the linear nature of PI control, homomorphic encryption such as Pallier encryption can be used to calculate the applied control while remaining encrypted, enhancing cybersecurity of the feedback control layer. Two nonlinear chemical process applications, including a benchmark chemical reactor example and one application modeled through the use of Aspen Dynamics, were used to demonstrate the application, and evaluate the performance and robustness of the proposed two-layer control architecture.

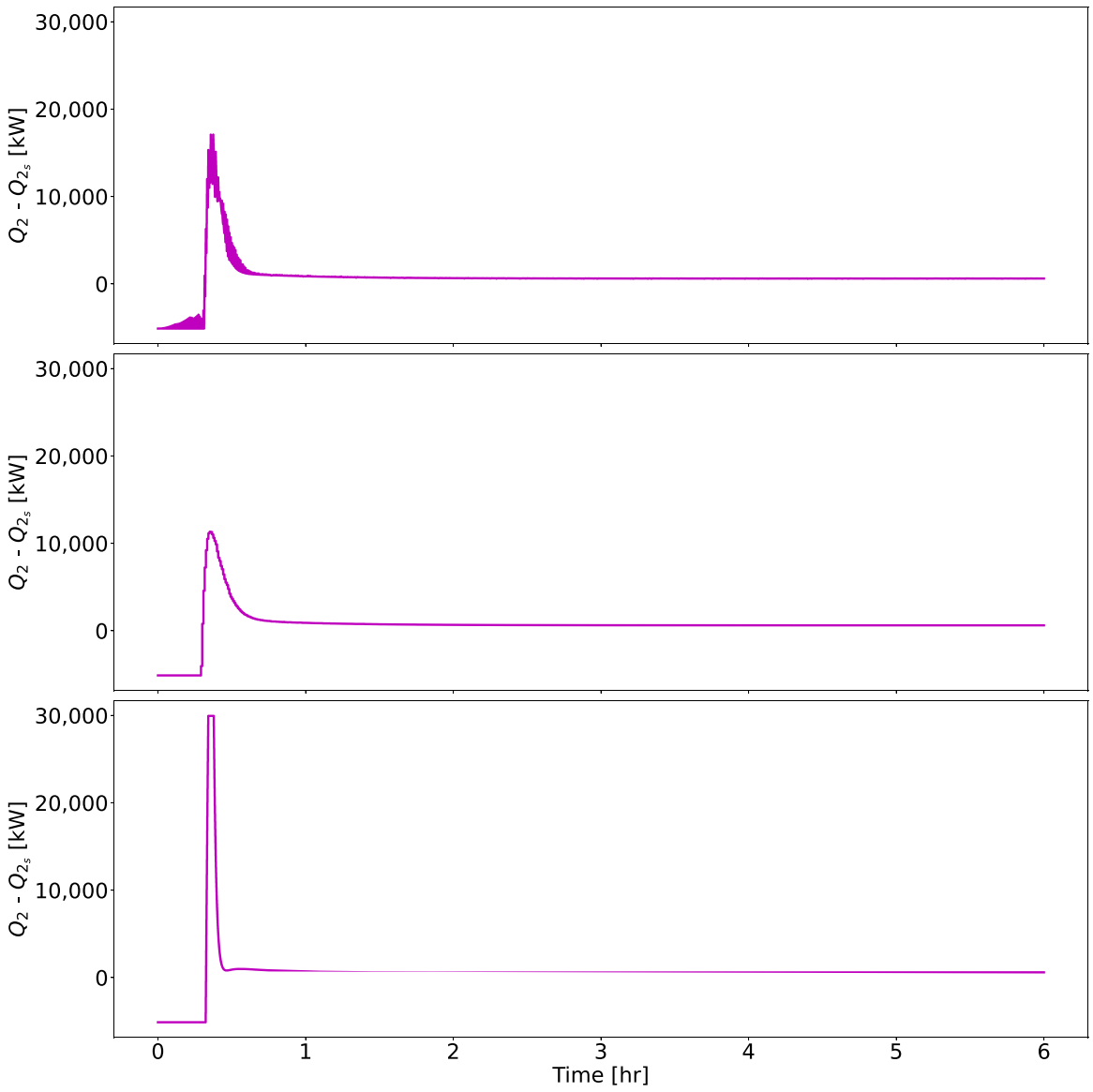

Fig. 13. Manipulated input profiles for the second CSTR control action (the top plot is the input under two-layer control, the middle plot is the input under tracking MPC and the bottom plot is the input under PI control only).

CRediT authorship contribution statement

Arthur Khodaverdian: Writing – original draft, Methodology, Investigation, Conceptualization. **Dhruv Gohil:** Writing – original draft, Methodology, Investigation, Conceptualization. **Panagiotis D. Christofides:** Writing – review & editing, Supervision, Methodology, Investigation, 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 the National Science Foundation, United States, CBET-2227241, is gratefully acknowledged.

References

(ACSC), A.C.S.C., 2024. Principles of Operational Technology Cyber Security. Technical Report, Australian Cyber Security Centre.

Aldrini, J., Chihi, I., Sidhom, L., 2024. Fault diagnosis and self-healing for smart manufacturing: a review. J. Intell. Manuf. 35, 2441–2473.

Aljohani, A., AlMuhaini, M., Poor, H.V., Binqadhi, H.M., 2024. A deep learning-based cyber intrusion detection and mitigation system for smart grids. IEEE Trans. Artif. Intell. 5, 3902–3914.

Babu, B., Ijyas, T., P., M., Varghese, J., 2017. Security issues in SCADA based industrial control systems. In: 2nd International Conference on Anti-Cyber Crimes. ICACC, Abha, Saudi Arabia, pp. 47–51.

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, L., Ye, Y., Bourlai, T., 2017. Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In: European Intelligence and Security Informatics Conference. Athens, Greece, pp. 99–106.

Conklin, W.A., 2016. IT vs. OT security: A time to consider a change in CIA to include resilienc. In: 49th Hawaii International Conference on System Sciences. Kauai, Hawaii, pp. 2642–2647.

Data61, C., 2013. Python paillier library.

Farwell, J.P., Rohozinski, R., 2011. Stuxnet and the future of cyber war. Survival 53, 23–40.

Ferraiolo, H., Regenscheid, A., 2024. Cryptographic Algorithms and Key Sizes for Personal Identity Verification. Technical Report NIST SP 800-78-5, National Institute of Standards and Technology, Gaithersburg, MD.

Gandhi, R., Sharma, A., Mahoney, W., Sousan, W., Zhu, Q., Laplante, P., 2011. Dimensions of cyber-attacks: Cultural, social, economic, and political. IEEE Technol. Soc. Mag. 30, 28–38.

Geng, H., Yang, G., 2014. Linear and nonlinear schemes applied to pitch control of wind turbines. Sci. World J. 2014, 406382.

Huang, H., Wlazlo, P., Mao, Z., Sahu, A., Davis, K., Goulart, A., Zonouz, S., Davis, C.M., 2022. Cyberattack defense with cyber-physical alert and control logic in industrial controllers. IEEE Trans. Ind. Appl. 58, 5921–5934.

Intel, 2024. Intel®64 and 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., 2024. Integrating dynamic economic optimization and encrypted control for cyber-resilient operation of nonlinear processes. AIChE J. 70, e18509.

Kadakia, Y.A., Alnajdi, A., Abdullah, F., Christofides, P.D., 2023. Encrypted distributed model predictive control with state estimation for nonlinear processes. Digit. Chem. Eng. 9, 100133.

Koay, A.M.Y., Ko, R.K.L., Hettema, H., Radke, K., 2023. Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges. J. Intell. Inf. Syst. 60, 377–405.

mechmotum, 2018. cyipopt. https://github.com/mechmotum/cyipopt.

Microsoft Security Team, 2024. Storm-0501: Ransomware attacks expanding to hybrid cloud environments. Microsoft Secur. Blog.

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: Stern, J. (Ed.), Advances in Cryptology — EUROCRYPT '99. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 223–238.

Pan, J., Sui, T., Liu, W., Wang, J., Kong, L., Zhao, Y., 2023a. Secure control using homomorphic encryption and efficiency analysis. Secur. Commun. Netw. 2023.

Pan, J., Sui, T., Liu, W., Wang, J., Kong, L., Zhao, Y., Wei, Z., 2023b. Secure control of linear controllers using fully homomorphic encryption. Appl. Sci. 13.

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. Proc. IEEE 106, 113–128.

Schulze Darup, M., Redder, A., Shames, I., Farokhi, F., Quevedo, D., 2018. Towards encrypted MPC for linear constrained systems. IEEE Control. Syst. Lett. 2, 195–200.

Sen, J., 2013. Homomorphic encryption — Theory and application. In: Sen, J. (Ed.), Theory and Practice of Cryptography and Network Security Protocols and Technologies. IntechOpen, Rijeka, http://dx.doi.org/10.5772/56687.

Simmons, C., Shiva, S., Dasgupta, D., Wu, C., 2009. AVOIDIT: A Cyber Attack Taxonomy. Technical Report, University of Memphis.

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, 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.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17, 261–272.

Wang, W., Harrou, F., Bouyeddou, B., Senouci, S.-M., Sun, Y., 2022. Cyber-attacks detection in industrial systems using artificial intelligence-driven methods. Int. J. Crit. Infrastruct. Prot. 38, 100542.

Wu, Z., Chen, S., Rincon, D., Christofides, P.D., 2020. Post cyber-attack state reconstruction for nonlinear processes using machine learning. Chem. Eng. Res. Des. 159, 248–261.