Mathematical Modelling for Secure Model Predictive Control in Process Engineering

Maithri Suresh (210020072)

Abstract

In recent years, cyber-security of networked control systems has become crucial, as these systems are vulnerable to targeted cyberattacks that compromise the stability, integrity, and safety of these systems. This paper proposes the establishment of secure and private communication links between sensor–controller and controller–actuator elements using semi-homomorphic encryption to ensure cyber security in model predictive control (MPC) of nonlinear systems. With this work as an example, I discuss the relevance of theory of mathematical modelling in solving complex real-life problems. I also implement the algorithm described in the paper (code files in this repository) and present the results.

Keywords
mathematical modelling, cyber security, encrypted control, model predictive control (MPC), quantization, semihomomorphic encryption

Introduction

Integration of cyber-secure strategies in physical networked control systems, to ensure secure and safe operation, has become crucial due to increased threats of targeted cyberattacks. Process engineering increasingly relies on data-driven models and predictive control strategies, which allow for optimal decision-making and real-time adjustments in complex systems. However, as these systems become interconnected via networks, they are exposed to cyber threats which can compromise data integrity and system stability. To address these risks, [1] proposes an encrypted model predictive control (MPC) framework that uses semi-homomorphic encryption — specifically the Paillier cryptosystem — to secure communication links between sensors, controllers, and actuators. This approach aims to protect data integrity and confidentiality by encrypting signals within the control loop, allowing the MPC to operate securely even in the presence of potential cyber threats. In this project, the algorithm and results of [1] are discussed. The parts relevant to mathematical modelling in process engineering are highlighted, while the other details are left for the appendix.

Proposed Encrypted MPC Design and Algorithm

For the proposed design shown in 1, signals $x(t)$ from the sensor are encrypted and sent to the model predictive controller (MPC). Before nonlinear computations are performed, the encrypted data is decrypted to obtain quantized states $\hat{x}(t)$ which are used to initialize the plant model in the MPC at time t. MPC calculates the optimized inputs $u(t)$, and these inputs are encrypted before being sent to the actuator. These encrypted inputs are further decrypted and the quantized inputs $\hat{u}(t)$ are applied to the process.

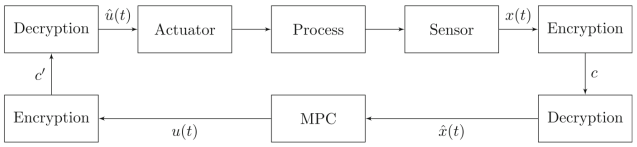

Figure 1. Schematic of closed-loop system under encrypted Model Predictive Control. [1]

Mathematically, the algorithm can be modelled using the following equations. For more on Paillier cryptosystem, Lyapunov functions and quantization please refer to the Appendix section.

System Dynamics

The continuous-time nonlinear system is described by:

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

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the manipulated input vector, and $f(x)$, $g(x)$ are sufficiently smooth functions.

Paillier Encryption of State and Input Data

The Paillier cryptosystem encrypts quantized messages $m \in \mathbb{Z}_M$ as follows:

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

where $r \in \mathbb{Z}_M$ is a random integer and c is the resulting ciphertext.

Quantization Mapping

Since Paillier encryption operates on integers, real number data is quantized. A real number a is mapped to a rational number q in a discrete set $Q_{l_1,d}$ through:

$$g_{l_1,d}(a) = \arg\min_{q \in Q_{l_1,d}} |a - q| \tag{3}$$

where $Q_{l_1,d}$ contains quantized values based on parameters l_1 and d.

Decryption of Encrypted Messages

The ciphertext $c \in \mathbb{Z}_{M^2}$ is decrypted using the private key to retrieve the original message:

$$D_M(c) = m = L(c^{\lambda} \mod M^2) \cdot u \mod M \tag{4}$$

where $L(x) = \frac{x-1}{M}$ and u is the modular inverse calculated during key generation.

Lyapunov-based Controller Stability Condition

A Lyapunov function $V(x)$ is used to ensure stability, satisfying:

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2 \quad \text{and} \quad \frac{\partial V(x)}{\partial x} F(x, \Phi(x)) \leq -c_3|x|^2 \tag{5}$$

where c_1, c_2, c_3 are positive constants, and $\Phi(x)$ represents the stabilizing control law.

MPC Cost Function Optimization

The MPC solves the following optimization problem:

$$J = \min_{u \in S(\Delta)} \int_{t_k}^{t_k+N} L(\hat{x}(t), u(t))\, dt \tag{6}$$

subject to constraints on the predicted state $\hat{x}(t)$, control inputs $u(t)$, and Lyapunov stability conditions.

Controller Parameter Switching for Cyber Security

The controller periodically switches parameters to detect cyberattacks, maintaining stability and ensuring encrypted data security.

Error Bounding in Quantization

Quantization errors are bounded as:

$$|x(t) - \tilde{x}(t)| \leq \eta_1 2^{-d} \quad \text{and} \quad |u(t) - \tilde{u}(t)| \leq \eta_2 2^{-d} \tag{7}$$

where η_1 and η_2 are constants related to the quantization parameters.

Closed-Loop Stability with Encrypted Control

For practical stability under encrypted control, the state $x(t)$ of the system remains in a defined region Ω_ρ, satisfying:

$$\dot{V} \leq -c_5|x|^2, \quad \limsup_{t \to \infty} |x| \leq b \tag{8}$$

where b depends on the quantization parameter d.

Example System: CSTR with Recycle

To apply the above algorithm and present results, [1] uses an example of a Continuous Stirred Tank Reactor (CSTR) system with recycle having the following systemic specifications and equations.

System Overview

The CSTR conducts an irreversible, second-order, exothermic reaction $A \to B$ within a well-mixed, non-isothermal setup. The system has a recycle stream and is equipped with a jacket for heat removal at a rate Q.

Material Balance

The differential equation describing the concentration C_A of reactant A is given by:

$$\frac{dC_A}{dt} = \frac{(1-\lambda)F}{V} C_A + \frac{\lambda F}{V} C_{A0} - \frac{F}{V} C_A - k_0 e^{-\frac{E}{RT}} C_A^2 \tag{9}$$

where:

- C_A is the concentration of reactant A in the reactor.
- C_{A0} is the inlet concentration of A.
- λ is the recycle fraction of the outlet stream, where λF is the product stream and $(1-\lambda)F$ is recycled back.
- k_0 is the pre-exponential factor, E is the activation energy, R is the gas constant, and T is the reactor temperature.

Energy Balance

The differential equation describing the temperature T in the reactor is:

$$\frac{dT}{dt} = \frac{(1-\lambda)F}{V} T + \frac{\lambda F}{V} T_0 - \frac{\Delta H}{\rho_L C_p} k_0 e^{-\frac{E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{10}$$

where:

- T_0 is the inlet temperature.
- ΔH is the heat of reaction.
- ρ is the density, C_p is the heat capacity, and V is the reactor volume.
- Q is the rate of heat removal.

Control Inputs and Constraints

The controlled inputs are C_{A0} (inlet concentration of A) and Q (heat removal rate), constrained within the ranges:

$$Q \in [-80.0, 80.0]\,\text{MJ/h}, \quad C_{A0} \in [0.5, 7.5]\,\text{kmol/m}^3$$

The CSTR is operated at a stable steady state:

$$[C_{A_s}, T_s] = [2.96\,\text{kmol/m}^3, 320\,\text{K}]$$

with control values:

$$Q_s = 12.2\,\text{MJ/h}, \quad C_{A0_s} = 4.0\,\text{kmol/m}^3$$

Quantization

For security, it is essential to quantize both the state and input data before encryption. Quantization is done using a function $gl_{1,d}(a)$ that maps real values to a discrete set $Q_{l_{1,d}}$, where l_1 controls the largest representable value, and d specifies the resolution. The maximum value in $Q_{l_{1,d}}$ is $2^{l_1-d-1} \cdot 2^{-d}$ which must be set to exceed permissible state and input values. Conversely, the minimum value is -2^{l_1-d-1} ensuring coverage of the operating range.

Quantization introduces a trade-off: higher values of d reduce quantization errors, bringing the quantized values closer to their real counterparts, but they also increase computational costs. For instance, increasing d improves the accuracy of calculations within the encrypted MPC framework but demands additional processing resources.

Lyapunov Function

The Lyapunov function for this example is constructed using a quadratic form, specifically $V = x^T Px$, where x represents the state vector in terms of deviations from the setpoints of concentration and temperature in the reactor, and P is a positive definite matrix. Extensive simulations helped to identify a suitable P matrix for stability, set to:

$$P = \begin{pmatrix} 500 & 20 \\ 20 & 1 \end{pmatrix} \tag{11}$$

Simulation Results

Besides the results summarised in the paper (see section 4 of [1]) where we see that as the value of d increases the quantization error decreases, here are some additional results obtained on implementation of the algorithm (refer this repository).

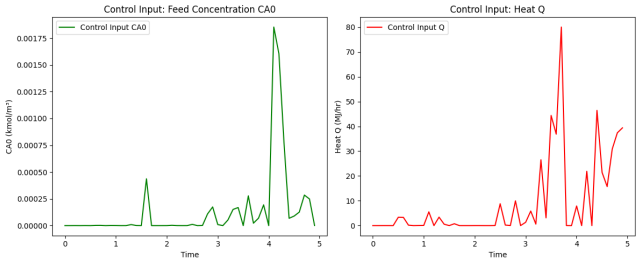

Figure 2. Control Inputs applied by the MPC over time for CSTR at Stable Steady State

Figure 2 shows the control inputs that the MPC applies over time to stabilize the system, while Figure 3 depicts how the concentration C_A and temperature T change over time as the MPC controls the system. Evidently, the concentration and temperature values approach a stable steady state as required.

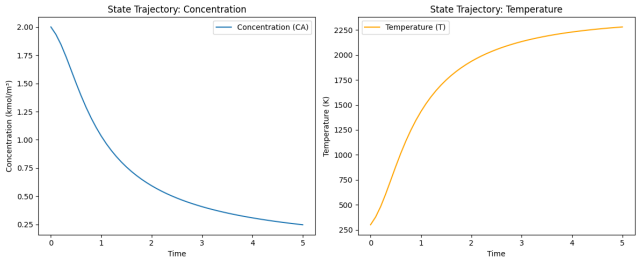

Figure 3. Response of State Variables to MPC action for CSTR at Stable Steady State

Example System: CSTR at Unstable Steady State

Another example is presented in [1] of a Continuous Stirred Tank Reactor (CSTR) system with the following systemic specifications and equations.

System Overview

The system is a jacketed, perfectly mixed CSTR conducting an irreversible, second-order, elementary, exothermic reaction $A \rightarrow B$. The system has a recycle stream and is equipped with a jacket for heat removal at a rate Q.

Material Balance

The differential equation describing the concentration C_A of reactant A is given by:

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

where:

- C_A is the concentration of reactant A in the reactor.
- C_{A0} is the inlet concentration of A.
- λ is the recycle fraction of the outlet stream, where λF is the product stream and $(1-\lambda)F$ is recycled back.
- k_0 is the pre-exponential factor, E is the activation energy, R is the gas constant, and T is the reactor temperature.

Energy Balance

The differential equation describing the temperature T in the reactor is:

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) - \frac{\Delta H}{\rho_L C_p} k_0 e^{-\frac{E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{13}$$

where:

- T_0 is the inlet temperature.
- ΔH is the heat of reaction.
- ρ is the density, C_p is the heat capacity, and V is the reactor volume.
- Q is the rate of heat removal.

Control Inputs and Constraints

The controlled inputs are C_{A0} (inlet concentration of A) and Q (heat removal rate), constrained within the ranges:

$$Q \in [-80.0, 80.0]\,\text{MJ/h}, \quad C_{A0} \in [0.5, 7.5]\,\text{kmol/m}^3$$

The CSTR is operated at an unstable steady state:

$$[C_{A_s}, T_s] = [1.95\,\text{kmol/m}^3, 402\,\text{K}]$$

with control values:

$$Q_s = 0\,\text{MJ/h}, \quad C_{A0_s} = 4.0\,\text{kmol/m}^3$$

Quantization

The logic is as detailed in the previous example.

Lyapunov Function

The Lyapunov function for this example is also constructed using a quadratic form, specifically $V = x^T P x$. Extensive simulations helped to identify a suitable P matrix for stability, set to:

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

The design also incorporates a stability criterion for the encrypted Lyapunov-based Model Predictive Control (LMPC), where a contractive constraint of $\dot{V} \leq -kV$ (with $k = 0.15$) ensures that the system's states remain within a stable region over the prediction horizon. This approach allows the LMPC to operate the reactor near its unstable steady state by stabilizing the state trajectories within predefined bounds.

Simulation Results

Besides the results summarised in the paper (see section 5 of [1]) where we see that as the value of d increases the quantization error decreases, here are some additional results obtained on implementation of the algorithm for this example (refer this repository).

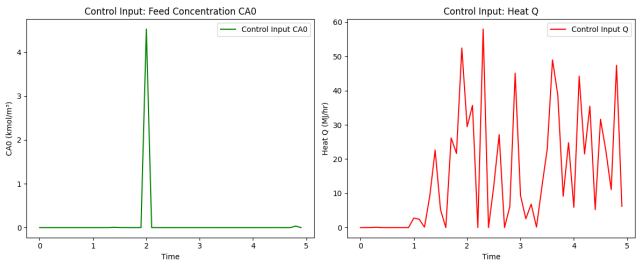

Figure 4. Control Inputs applied by the MPC over time for CSTR at Unstable Steady State

Figure 4 shows the control inputs that the MPC applies over time to stabilize the system, while Figure 5 depicts how the concentration C_A and temperature T change over time as the MPC controls the system. Evidently, the concentration and temperature values approach a steady state, but much slower than the previous example system. Additionally, the control inputs, specifically heat, shows a high variance implying unsteadiness of the system.

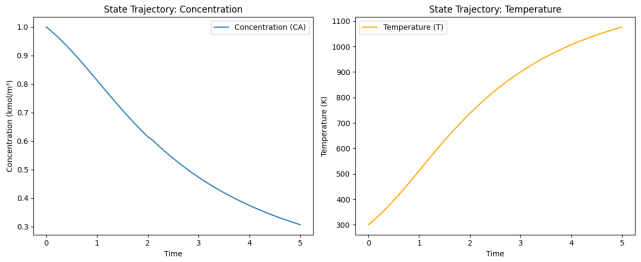

Figure 5. Response of State Variables to MPC action for CSTR at Unstable Steady State

Conclusion

Thus, through mathematical modelling, the paper [1] develops a secure control framework by combining Model Predictive Control with the Paillier encryption scheme where stability and control performance are maintained despite encryption-related quantization errors to protect control actions and state data from cyber threats. The role of advanced mathematical techniques is critical in creating resilient and secure control architectures that are essential as process engineering increasingly adopts interconnected and data-driven systems.

Acknowledgement

I extend my heartfelt gratitude towards my course instructor and project mentor, Prof. Santosh Noronha, for his continuous guidance throughout the course and this project. His high standards of work and clarity of thought have been instrumental in my overall understanding of the course and pushed me to peak performance. His unwavering work ethic pushed me to maintain a deep sense of rigour in my academic pursuits.

References

[1] Suryavanshi A, Alnajdi A, Alhajeri M, Abdullah F, Christofides PD. Encrypted model predictive control design for security to cyberattacks. AIChE J. 2023; 69(8):e18104. doi:10.1002/aic.18104

[2] Paillier, P. (1999). Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (eds) Advances in Cryptology — EUROCRYPT '99. EUROCRYPT 1999. Lecture Notes in Computer Science, vol 1592. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48910-X_16

[3] Lecture 12 - Basic Lyapunov Theory

[4] Lyapunov Functions and Interaction Analysis and Multiloop Control

[5] What is Quantization?

Appendix

Class of Systems [1]

In this work, the focus on continuous-time nonlinear systems of nonlinear first-order ordinary differential equations (ODEs) with inputs of the form:

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

where $x = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^m$ is the manipulated input vector. The inputs to the process are bounded, that is, $u \in U$ where the set $U \in \mathbb{R}^m$ is defined as $U := \{u \in U | u_{min,i} \leq u_i \leq u_{max,i}, \forall i = 1,2,...,m\}$. $u_{min,i}$ and $u_{max,i}$ are physical bounds and define the minimum and maximum value that each manipulated input can attain. $f(\cdot)$ is a sufficiently smooth vector function and $g(\cdot)$ is a sufficiently smooth matrix function. Without loss of generality, it is assumed $f(0) = 0$ and, hence, the origin is a steady state of the above nonlinear system.

Paillier Cryptosystem [2]

The Paillier cryptosystem is a probabilistic public-key encryption scheme with homomorphic properties, allowing certain operations on encrypted data. Mathematically, it is based on the difficulty of factoring large integers and uses modular arithmetic for encryption and decryption.

Key Generation

1. *Choose Two Large Primes*: Select two large prime numbers, p and q, such that $p \neq q$.

2. *Compute Modulus n*: Compute $n = p \cdot q$. This integer n will serve as the modulus for the encryption and decryption operations.

3. *Compute λ*: Define λ as the least common multiple of $p-1$ and $q-1$:

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

4. *Choose g*: Select g such that g is in $\mathbb{Z}^*_{n^2}$ (the multiplicative group modulo n^2), and typically, $g = n+1$ is chosen as it simplifies calculations.

5. *Compute μ*: Compute μ as the modular inverse of $L(g^\lambda \mod n^2)$ modulo n, where L is defined as:

$$L(u) = \frac{u-1}{n}.$$

Thus, $\mu = \left(L(g^\lambda \mod n^2)\right)^{-1} \mod n$.

6. *Public and Private Keys*:

   - The public key is (n, g).
   - The private key is (λ, μ).

Encryption

Given a plaintext message m where $m \in \mathbb{Z}_n$, encryption proceeds as follows:

1. *Choose a Random r*: Select a random integer r such that $r \in \mathbb{Z}^*_n$ (i.e., r is coprime to n).

2. *Compute Ciphertext c*: Encrypt m by computing

$$c = g^m \cdot r^n \mod n^2.$$

The randomness introduced by r ensures that encrypting the same m multiple times yields different ciphertexts, making the scheme probabilistic.

Decryption

Given a ciphertext c, decrypt it as follows:

1. *Compute Intermediate Value*: Calculate $u = c^\lambda \mod n^2$.

2. *Apply L Function*: Compute $L(u) = \frac{u-1}{n}$.

3. *Recover Plaintext m*: Multiply $L(u)$ by μ modulo n to obtain the plaintext:

$$m = (L(c^\lambda \mod n^2) \cdot \mu) \mod n.$$

The Paillier cryptosystem is additively homomorphic, meaning that the product of two ciphertexts decrypts to the sum of their respective plaintexts:

1. Given two plaintexts m_1 and m_2 with ciphertexts $c_1 = g^{m_1} r_1^n \mod n^2$ and $c_2 = g^{m_2} r_2^n \mod n^2$,

2. The product $c = c_1 \cdot c_2 \mod n^2$ is a valid encryption of $m_1 + m_2 \mod n$.

Lyapunov Functions [3, 4]

A Lyapunov function is a scalar function $V(x)$ used in control theory to analyze the stability of an equilibrium point of a dynamical system. For a dynamical system described by

$$\dot{x} = f(x),$$

an equilibrium point x_e (often $x_e = 0$) is said to be Lyapunov stable if there exists a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ satisfying the following properties:

1. Positivity: $V(x) > 0$ for $x \neq x_e$ and $V(x_e) = 0$.

2. Radial Unboundedness (often required in some cases): $V(x) \to \infty$ as $\|x\| \to \infty$.

3. Negative Semi-Definiteness of the Time Derivative: The derivative of $V(x)$ along the trajectories of the system, denoted $\dot{V}(x) = \frac{d}{dt}V(x) = \nabla V \cdot f(x)$, is non-positive, $\dot{V}(x) \leq 0$, for all x.

If these conditions hold, $V(x)$ is called a Lyapunov function for the system, and it guarantees that x_e is stable in the sense of Lyapunov.

If $\dot{V}(x) < 0$ for all $x \neq x_e$, then x_e is asymptotically stable, meaning that trajectories starting close to x_e not only remain close but eventually converge to x_e.

Quantization [5]

Quantization is defined as a function Q that maps a value x from a continuous set $\mathbb{R}$ to a discrete set $\{q_1, q_2, \ldots, q_n\}$, often called quantization levels. Formally:

$$Q : \mathbb{R} \to \{q_1, q_2, \ldots, q_n\}.$$

For any input $x \in \mathbb{R}$, $Q(x)$ is the quantized (or approximated) value.

Quantization can be uniform or non-uniform.

1. *Uniform Quantization*: In uniform quantization, the quantization levels are evenly spaced. For a uniform quantizer with step size Δ, any value x is mapped to the nearest quantization level q_i such that:

   $$Q(x) = \Delta \cdot \text{round}\left(\frac{x}{\Delta}\right).$$

   This is commonly used in digital audio and video encoding because it simplifies implementation and ensures consistent error across the range.

2. *Non-Uniform Quantization*: In non-uniform quantization, the quantization levels are not evenly spaced. This type is often used when values are more densely distributed in certain ranges (e.g., human speech frequencies in audio encoding). Non-uniform quantization can be designed to reduce error for more common values.

Quantization Error

Quantization introduces an error because continuous values are approximated by discrete levels. The quantization error e for a value x is defined as the difference between the original value and the quantized value:

$$e = x - Q(x).$$

For a uniform quantizer with step size Δ, the quantization error is bounded by $-\frac{\Delta}{2} \leq e < \frac{\Delta}{2}$.

To summarise, quantization refers to the process of mapping a large (or continuous) set of values to a smaller (or discrete) set. This concept is fundamental in fields like signal processing, control theory, and quantum mechanics, where it often involves converting a continuous signal or variable into a discrete representation. Quantization can be thought of as a form of approximation that reduces the amount of information by grouping values together, which is useful when working within limited storage or processing constraints.

Graphical results regarding quantization from [1]

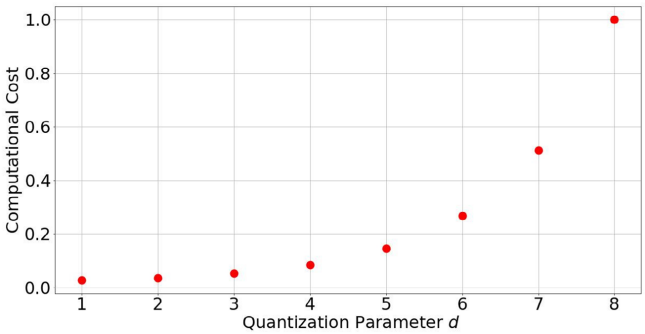

Figure 6. Normalized computational cost associated with different values of the quantization parameter d.

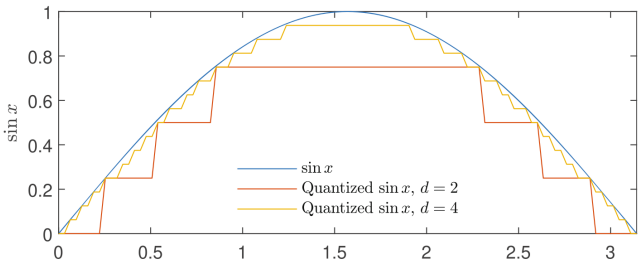

Figure 7. Demonstration of the effect of varying d on the quantization error for the sine function.