Encrypted Control for Flat Nonlinear Cyber-Physical Systems

Timothée Schmoderer, Estelle Courtial and Matthieu Fruchard

***Abstract*— This paper deals with the design of an encrypted control for a class of nonlinear cyber-physical systems. A fully homomorphic encryption algorithm performing on real numbers is used to encrypt the measured outputs and to perform computations on encrypted data. We get around the problem of nonlinearity by exploiting the intrinsic property of a class of nonlinear systems, namely the differential flatness. A Luenberger observer based on the linear equivalent Brunovski form of the nonlinear flat system allows to reconstruct the state vector, necessary to compute a stabilising state feedback control. Simulations on a mobile robot illustrate the theoretical results, ensuring a secure way to control flat nonlinear cyber-physical systems.**

***Index Terms*— Nonlinear systems, encrypted control, flatness, cyber-physical systems, fully homomorphic encryption.**

I. Introduction

Networked Control systems (NCS) and cyber-physical systems (CPS) are largely widespread both in industry 4.0 and in transportation, energy or medical infrastructures. These systems are composed of several physical and software entities that exchange data. Data processing and/or computations (e.g., computation of control inputs or estimated states) are often outsourced to a cloud-server due to the limitations of computational resources embedded on the plant or to the need of a centralized supervision. As soon as there is an exchange of information (e.g., via a network or a wireless technology), these systems are vulnerable to cyber-attacks. Different types of cyber-threats and cyber-attacks such as denial-of-service, eavesdropping, remote control attack, are reported in the literature [1], [2]. To face these threats and attacks, a holistic approach is to combine a resilient strategy and a strategy that mitigates the threats. The last objective can be addressed by the data encryption in order to protect data in terms of confidentiality, integrity and availability. In 2009, Gentry [3] developed a fully homomorphic encryption (HE) scheme allowing operations (multiplication and addition) on encrypted data without decryption. Few years later, the control community took advantage of the HE for control purposes. Kogiso et al. [4] in 2015 proposed the first encrypted controller based on an output feedback for a linear system with the RSA encryption algorithm. Since then, numerous works (the reader can refer to the complete survey by Schlüter et al. [5] and references therein, [6], [7]) have been published, using either partial HE algorithms (first generation of encrypted controller) or full HE (FHE) algorithms (second generation of encrypted controller). This work falls into the second generation and attempts to meet some of the remaining challenging issues in encrypted control. First, the majority of the works deals with linear systems [6], [8], [9] and only few works concern nonlinear systems [10]–[13], mainly considering Stone-Weierstrass approximation. Secondly, most of the encryption algorithms can only perform on integers while dynamic systems use real numbers. Finally, the control strategy is always developed in discrete-time while dynamical plants are continuous-time.

To address theses issues, the main contributions of this paper are : i) we consider nonlinear systems and exploit the flatness property to design an encrypted control based on the equivalent Brunovski form, ii) we use the CKKS [14], a fully homomorphic encryption algorithm, which can be applied on real numbers contrary to the usual algorithms (RSA, Pallier and LWE) devoted to integers, iii) we develop a continuous-time encrypted control: a Luenberger observer provides the encrypted estimated states necessary for the observer-based feedback control strategy.

Simulations are carried out on a mobile robot to highlight the efficiency of the proposed control strategy and offer an insight on our theoretical results for flat nonlinear CPS.

The remaining of the paper is organised as follows. The CKKS encryption algorithm is presented in Section II. The encrypted control architecture is described and detailed in Section III as well as the synthesis of the state-feedback and the stability study. Section IV shows a numerical example and Section V finally concludes this paper.

Notations: Through this paper, $\mathbb{R}$ is the set of real numbers, $\mathbb{Z}_q$ is the set of integers modulo $q \in \mathbb{N}$. The symbol $\lfloor . \rceil$ denotes the rounding to the nearest integer function. A vector is noted in bold $\mathbf{v}$ and v_i is the $i-th$ component of the vector. The plaintext (clear message) is the input of the encryption algorithm which provides the unreadable output, called ciphertext.

II. Encryption algorithm: CKKS

The Cheon-Kim-Kim-Song (CKKS) fully homomorphic encryption scheme [14] allows to perform homomorphic computations over real numbers. This procedure supports approximate arithmetic over complex numbers (hence, real numbers) with a level of error comparable to the one made by floating point arithmetic on traditional computers. The CKKS framework consists of five main algorithms: encryption, decryption, addition, multiplication, and rescaling. The first two algorithms transform plaintexts (complex vectors) into ciphertexts (polynomial elements) and vice-versa; the next

This work was not supported by any organization

All authors are with University of Orléans, {timothee.schmoderer, estelle.courtial, matthieu.fruchard}@univ-orleans.fr

two perform homomorphic operations on encrypted data; the rescaling procedure is used to maintain the size of the message and to control the growth of the error after homomorphic operations. A detailed presentation of the CKKS scheme is out of the scope of this paper, we refer the interested reader to [14] and [15]. Nevertheless, in the sequel, we focus on the intuition behind the encryption/decryption procedure and the evaluation of homomorphic operations. We summarize the main properties of the CKKS scheme in Lemma 1.

a) CKKS parameters: The CKKS scheme uses the following parameters: N is a power of two giving the dimension of the ciphertext domain; $q_L = \Delta^L q_0$ is the modulus of the space, where Δ is the scaling factor, L is an integer chosen according to the depth of the evaluation circuit, and q_0 is the basis modulus. Moreover, the polynomial ring space $\mathcal{R}_q$ is defined by $\mathbb{Z}_q[X]/\left(X^N+1\right)$. Elements of $\mathcal{R}_q$ are polynomials with integer coefficients bounded by q.

b) Encryption/Decryption: In the CKKS encryption procedure, the first step is an *encoding* step: the clear text message $\mathbf{m} \in \mathbb{C}^{N/2}$ is transformed into an element of the polynomial ring $\mathcal{R}_q$. This operation is completely transparent, thus, without loss of generality, we assume that the plaintexts are already polynomials. The encryption/decryption procedures rely on the generation of a public key and a secret key. The secret key $\mathtt{SK}$ is used for the decryption (it can also be used for encryption in a symmetric encryption scheme framework) and is taken as a random polynomial of degree N with coefficients in $\{-1,0,1\}$. The public key is a pair of polynomials $(\mathtt{PK}_1, \mathtt{PK}_2)$ given by:

$$\mathtt{PK}_1 = -a \cdot \mathtt{SK} + e \mod q_L \quad \text{and} \quad \mathtt{PK}_2 = a, \tag{1}$$

where a is a random polynomial sampled uniformly from $\mathcal{R}_{q_L}$ and e is a random error polynomial sampled from the discrete Gaussian distribution with variance σ^2. To encrypt a plaintext $\mathbf{m}$, we generate three small random polynomials μ with coefficients in $\{-1,0,1\}$, e_1 and e_2 from the discrete Gaussian distribution with variance σ^2, and we construct the ciphertext $\mathsf{Enc}(\mathbf{m}) = \mathbf{c} \in \mathcal{R}_{q_L}^2$ given by the pair (c_1, c_2) defined as follows:

$$c_1 = \mathtt{PK}_1 \cdot \mu + e_1 + \lfloor \Delta \mathbf{m} \rceil \mod q_L \tag{2}$$

$$c_2 = \mathtt{PK}_2 \cdot \mu + e_2 \mod q_L. \tag{3}$$

The encryption process injects an error in the message. Hence, to preserve the precision of the message, we multiply it by a scaling factor $\Delta > 1$.
The decryption is performed by evaluating the ciphertext on the secret key to generate an approximate value $\mathbf{m}'$ of the plaintext $\mathbf{m}$:

$$\mathbf{m}' = \mathsf{Dec}(\mathbf{c}) = \frac{1}{\Delta}\left(c_1 + c_2 \cdot \mathtt{SK} \mod q_L\right). \tag{4}$$

We say that a ciphertext $\mathbf{c}$ contains a noise e if the decrypted value $\mathbf{m}'$ differs from the exact plaintext $\mathbf{m}$ by e.

c) Homomorphic operations: The main advantage of using a FHE scheme is to perform arithmetic operations in the cipher domain that are *equivalent* to the operations in the clear domain. With CKKS, the addition is straightforward: we simply add the corresponding polynomials of each ciphertext. Consider two ciphertexts $\mathbf{c}^1 = (c_1^1, c_2^1)$ and $\mathbf{c}^2 = (c_1^2, c_2^2)$, the addition is defined as :

$$\mathbf{c}^1 \oplus \mathbf{c}^2 = \left(c_1^1 + c_1^2, c_2^1 + c_2^2\right) \mod q_L. \tag{5}$$

The multiplication consists in two steps: the multiplication of the polynomials and a rescaling step which aims at reducing the noise introduced by the first step. We set $(d_0, d_1, d_2) = (c_1^1 c_1^2, c_2^1 c_1^2 + c_1^2 c_2^1, c_2^1 c_2^2)$. The multiplication of two ciphertexts is:

$$\mathbf{c}^1 \otimes \mathbf{c}^2 = \left\lfloor \frac{1}{\Delta}\left((d_0, d_1) + \lfloor P^{-1} d_2 \mathtt{EK} \rceil \mod q_L\right) \right\rceil, \tag{6}$$

where $\mathtt{EK} = (-a' \cdot \mathtt{SK} + e' + P \cdot \mathtt{SK}^2 \mod P q_L, a')$ is the evaluation key given by an integer P, and two polynomials a' and e' constructed like for the public key (cf eq(1)).
The following lemma gives a bound on the error made when computing arithmetic operations with approximate arithmetic.

Lemma 1 ([14])**.** *Let $\mathbf{m}' = \mathsf{Dec}(\mathsf{Enc}(\mathbf{m}))$, then $\mathbf{m}' = \mathbf{m} + O(N/\Delta)$. Let $\mathbf{c}^i$, for $i = 1, 2$, be a ciphertext (with noise e_i bounded by $\|\epsilon_{ni}\|_\infty$) of the plaintext $\mathbf{m}_i$ satisfying $\|\mathbf{m}_i\|_\infty \leq \nu_i$. Then, the noise included in the ciphertext $\mathbf{c}_a = \mathbf{c}^1 \oplus \mathbf{c}^2$ is bounded by $\|\epsilon_{n1}\|_\infty + \|\epsilon_{n2}\|_\infty$ and in $\mathbf{c}_m = \mathbf{c}^1 \otimes \mathbf{c}^2$ is bounded by $\frac{1}{\Delta}\left(\nu_1\|\epsilon_{n2}\|_\infty + \nu_2\|\epsilon_{n1}\|_\infty + \|\epsilon_{n1}\|_\infty\|\epsilon_{n2}\|_\infty\right) + \epsilon_{CKKS}$, with $\epsilon_{CKKS} = O(N/\Delta)$.*

Notice, that the CKKS scheme performs only approximate arithmetic operations since the noise included (for security reasons) in the payload can not be separated from the original message, but this is similar as how approximate arithmetic works on computers.

Remark 1. It is also possible to carry out homomorphic operations between plaintext and ciphertext. In that case, the Lemma takes in account the noise bound $\|\epsilon_{ni}\|_\infty$ of the plaintext set to zero. On the one hand, by using this kind of operations, we lower the rate at which the noise grows in the ciphertext, but, on the other hand, some of the information will not be encrypted and will be accessible to cyber-attackers.

III. Synthesis of the encrypted control: closed-loop architecture and stability study

A. Brief recall on flatness

First introduced by [16], the differential flatness property of a class of nonlinear systems allows to formulate straightforward a trajectory planning and the corresponding control synthesis. Consider a nonlinear system described by:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \tag{7}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ and $\mathbf{u} \in \mathbb{R}^m$ are respectively the state vector and the control vector. The system (7) is said to be differentially flat if there exists a flat output $z = (z_1, \ldots, z_p) \in \mathbb{R}^p$ such that :

$$z(t) = h(\mathbf{x}(t), \mathbf{u}(t), \mathbf{u}^{(1)}(t), \ldots, \mathbf{u}^{(\alpha)}(t)) \quad (8)$$

$$\mathbf{x}(t) = \Phi(z(t), z^{(1)}(t), z^{(2)}(t), \ldots, z^{(\beta)}(t)) \quad (9)$$

$$\mathbf{u}(t) = \Psi(z(t), z^{(1)}(t), z^{(2)}(t), \ldots, z^{(\gamma)}(t)) \quad (10)$$

where h, Φ, Ψ are smooth functions and (α, β, γ) are multi-indices integers. There are two reasons to not use a bold notation for the flat outputs vector z: first, each flat output z_i can be treated independently of the others and second, we keep the bold notation for the state of the Brunovski system that encompasses z_i and its derivatives up to order $\gamma_i - 1$ (see below).

The use of flatness is twofold: on one hand, we can compute the control inputs based on the reference trajectory. Indeed, for a sufficiently smooth reference trajectory of the flat output $z_{ref}(t)$, equation (10) allows to compute (without integrating any differential equation) the corresponding open-loop control $\mathbf{u}_{ref}(t)$. On the other hand, we can exploit the equivalent linear normal form of the nonlinear system. Using the differential flatness to generate open-loop controls can be considered as a dynamic feedback linearization

$$\dot{\mathbf{w}} = a(\mathbf{x}, \mathbf{w}, \mathbf{v}) \quad (11)$$

$$\mathbf{u} = b(\mathbf{x}, \mathbf{w}, \mathbf{v}), \quad (12)$$

which yields a linear system in the Brunovski form for each flat output z_i given by :

$$\dot{\mathbf{z}}_i = A_i \mathbf{z}_i + B_i v_i, \quad (13)$$

where $\mathbf{z}_i = \left(z_i, \dot{z}_i, \ldots, z_i^{(\gamma_i - 1)}\right) \in \mathbb{R}^{\gamma_i}$, A_i is a zero matrix except for the upper diagonal which is composed of ones, $B_i = (0, \ldots, 0, 1)^t$, and v_i is the flat control. Note that the flat output z_i is observed from the above system via the relation $z_i = C_i \mathbf{z}_i$ with $C_i = (1, 0, \ldots, 0)$.

In the sequel, we suppose that the nonlinear system (7), modeling cyber-physical systems considered in this study, is differentially flat with the flat output $z(t)$.

B. Encrypted state-feedback control architecture

The control objective is to compute a control law $\mathbf{u}(t) \in \mathbb{R}^m$ such that the nonlinear output $z(t) \in \mathbb{R}^p$ of the system (7) tracks a given reference trajectory $z_{ref}(t)$. To do so, we exploit the flatness of the nonlinear system to address the tracking trajectory problem for each flat output z_i.

In the context of networked control systems or cyber-physical systems, we suppose that the control law calculation is delegated to a cloud server. Thus, to ensure the system security, we encrypt the measured output z and send it to the cloud service. Thanks to the FHE algorithm (CKKS algorithm), all the computations are performed over encrypted data, guaranteeing the data protection against eavesdropping for example. Finally, the encrypted control is sent to the CPS, which decrypts and applies it. The proposed architecture, dealing with the secure control of CPS or NCS is displayed in figure 1.

The implementation of such control strategy requires several steps such as state-observer and state-feedback synthesis, that we detail hereafter. Given the reference trajectory $z_{ref}(t)$, we compute its derivatives (up to order $\gamma - 1$) and thus, we obtain a reference trajectory $\mathbf{z}_{i,ref}$ for each state $\mathbf{z}_i$ of the Brunovski system (13). Moreover, for each flat output we get the corresponding reference control $v_{i,ref} = z_{i,ref}^{(\gamma)}$.

State-observer: a Luenberger observer is constructed for each flat output $z_i(t)$ (considered to be measured) to recover the derivatives $\dot{z}_i(t), \ldots, z_i^{(\gamma_i - 1)}(t)$:

$$\dot{\hat{\mathbf{z}}}_i = A_i \hat{\mathbf{z}}_i + B_i v_i + L_i C_i (\mathbf{z}_i - \hat{\mathbf{z}}_i). \quad (14)$$

The matrix L_i is chosen such that $A_i - L_i C_i$ is Hurwitz.

State-feedback: we first define the estimated tracking error by setting $\tilde{\hat{\mathbf{z}}} = \hat{\mathbf{z}} - \mathbf{z}_{ref}$ with $\hat{\mathbf{z}} = (\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_p)$ and $\tilde{\hat{\mathbf{z}}}$, $\mathbf{z}_{ref}$ with a similar notation. We compute a stabilising feedback

$$\mathbf{v} = \tilde{\mathbf{v}} + \mathbf{v}_{ref}, \quad (15)$$

with $\mathbf{v} = (v_1, \ldots, v_p)$ and $\tilde{\mathbf{v}}_i = -K_i \tilde{\hat{\mathbf{z}}}_i$, for all $1 \leq i \leq p$. The gain matrix K_i is chosen such that $A_i - B_i K_i$ is Hurwitz. Finally, we apply the stabilising control to the nonlinear system using equation (10) with $\hat{\mathbf{z}}$ and $\mathbf{v}$.

The security of the scheme is guaranteed by performing all operations in the encrypted domain using homomorphic operations. Namely, the following data are encrypted: the reference trajectory and its derivatives, the controls, the estimated states of the Brunovski system, and the matrices K_i and L_i. The integration of the observer (14) and the feedback control computation are consequently performed in the encrypted domain.

Remark 2. Notice that we do not need to encrypt the matrices A, B, and C, because they are only composed of zeros or ones. Thus, their role in (14) is completely transparent, highlighting another benefit of using the flatness property.

Remark 3. Equation (10) has to be evaluated in the clear domain because of the nonlinearity of Ψ. However, it is sometimes possible to obtain a relation between the control $\mathbf{u}$ of the nonlinear system and the control $\mathbf{v}$ that involves only additions and multiplications. In that case, the evaluation of Ψ can be done within the cloud service (in the encrypted domain) to relieve the charge on the physical system. In the numerical example below, such a scenario is illustrated.

C. Stability study

In the following theorem, we show that the error introduced by the encryption scheme does not significantly affect the stability of the closed-loop nonlinear system. In the following, all decrypted quantities are assigned with a prime superscript.

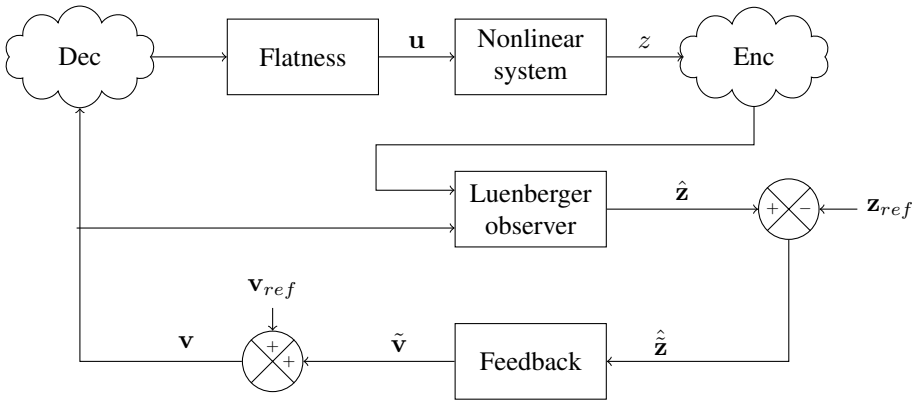


Fig. 1: Encrypted observer-based state-feedback control architecture for a flat nonlinear system.

Theorem 2. *Consider a nonlinear system* (7) *and assume that it is differentially flat, so it is mapped to the trivial system* (13)*. Let*

$$\begin{aligned}\tilde{\mathbf{v}}' &= -K'\tilde{\hat{\mathbf{z}}}' \\ \dot{\tilde{\hat{\mathbf{z}}}}' &= A\tilde{\hat{\mathbf{z}}}' + B\tilde{\mathbf{v}}' + L'C(\tilde{\mathbf{z}}' - \tilde{\hat{\mathbf{z}}}')\end{aligned} \tag{16}$$

be the encrypted controller and observer defined in the previous section. Noise, quantization and CKKS product induce errors, respectively denoted as ϵ_n, ϵ_q and ϵ_{CKKS}, which affect the realization of (14) and (15), so the prime variables differ from nominal ones. Then, the encrypted observer-based feedback control (16) *achieves the practical stabilization of the system* (7) *in a ball whose radius can be chosen arbitrarily small.*

Proof. Let us consider the equivalent flat model of (7), given by the Brunovski form

$$\Lambda \,:\, \dot{\tilde{\mathbf{z}}} = A\tilde{\mathbf{z}} + B\tilde{\mathbf{v}}.$$

Due to the linear canonical nature of this system, choosing K, L such that $A_e = \begin{pmatrix} A-BK & BK \\ 0 & A-LC \end{pmatrix}$ is Hurwitz implies the existence of P symmetric definite positive satisfying $A_e^t P + PA_e = -I$. Let $\mathbf{e} = (\tilde{\mathbf{z}}^t, \tilde{\hat{\mathbf{z}}}^t)^t$, if no disturbance affects the system, defining a Lyapunov function as $V = \mathbf{e}^t P \mathbf{e}$ yields:

$$\lambda_P^- \|\mathbf{e}\|^2 \leq V(\mathbf{e}) \leq \lambda_P^+ \|\mathbf{e}\|^2 \tag{17}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial \mathbf{e}}\dot{\mathbf{e}} \leq -\|\mathbf{e}\|^2 \tag{18}$$

$$\frac{\partial V}{\partial \mathbf{e}} \leq 2\lambda_P^+ \|\mathbf{e}\| \tag{19}$$

where $(\lambda_P^-, \lambda_P^+)$ are the minimal and maximal eigenvalues of P, so the undisturbed nominal system is globally exponentially stable.

However, the encryption/decryption procedure affects the observer and control computation since matrices K, L and any decrypted vector $\mathbf{d}$ are affected by noise and quantization errors resulting in $K' = K + \epsilon_K$, $L' + \epsilon_L$, $\mathbf{d}' = \mathbf{d} + \epsilon_d$ values where $\epsilon_\cdot = \epsilon_{n\cdot} + \epsilon_{q\cdot}$ for each scalar component. Due to flatness, matrices A, B, C entries are in $\{0, 1\}$ so no error is made at this level. Encryption/decryption affects the system when computing the control and observer as (16). Therefore we have the following error system:

$$\dot{\tilde{\mathbf{z}}} = A\tilde{\mathbf{z}} + B\tilde{\mathbf{v}} + g_1(\tilde{\hat{\mathbf{z}}}) \tag{20}$$

$$\dot{\tilde{\hat{\mathbf{z}}}} = A\tilde{\hat{\mathbf{z}}} + B\tilde{\mathbf{v}} + LC(\tilde{\mathbf{z}} - \tilde{\hat{\mathbf{z}}}) + g_2(\tilde{\hat{\mathbf{z}}}, \tilde{\mathbf{z}}) \tag{21}$$

with disturbances $g_1 = B(-K'\tilde{\hat{\mathbf{z}}}' + K\tilde{\hat{\mathbf{z}}})$ and $g_2 = A(\tilde{\hat{\mathbf{z}}}' - \tilde{\hat{\mathbf{z}}}) + g_1 + (L'C(\tilde{\mathbf{z}}' - C\tilde{\hat{\mathbf{z}}}') - LC(\tilde{\mathbf{z}} - \tilde{\hat{\mathbf{z}}}))$. Then, the following bounds can be derived: $\|g_i\|_2 \leq \delta_i = \alpha_i|\epsilon_n + \epsilon_q| + \beta_i \epsilon_{CKKS}$, where ϵ_{CKKS} is generated as soon as a product of encrypted quantities is performed (see lemma 1) and with

$$\alpha_1 = n\sqrt{m}(\|K\|_{max}\sqrt{2} + \sqrt{m}\|\tilde{\hat{\mathbf{z}}}\|_\infty) + \sqrt{m} \tag{22}$$

$$\beta_1 = \sqrt{nm} \tag{23}$$

$$\alpha_2 = \alpha_1 + p\sqrt{n}(\|L\|_{max}\sqrt{2} + \sqrt{n}\|\tilde{\hat{\mathbf{z}}}\|_\infty) + \sqrt{n} \tag{24}$$

$$\beta_2 = \beta_1 + \sqrt{np} \tag{25}$$

and $\delta(\delta_1, \delta_2)$ is an upper bound for the non vanishing disturbance g_e that affects the system (20)-(21). Using Lyapunov function $V = \mathbf{e}^T P \mathbf{e}$ along the trajectories of (20) and (21) results in:

$$\dot{V} \leq -\|\mathbf{e}\|^2 + 2\lambda_P^+ \delta \|\mathbf{e}\| \tag{26}$$

$$\leq -(1-\lambda)\|\mathbf{e}\|^2 - \|\mathbf{e}\|(\lambda\|\mathbf{e}\| - 2\lambda_P^+\delta), \tag{27}$$

for all $\lambda \in (0, 1)$. Thus, for any $\|\mathbf{e}\| \geq \frac{2\lambda_P^+\delta}{\lambda}$ we have $\dot{V} \leq 0$. Hence it follows from the global exponential stability in the nominal case that the disturbed system is uniformly ultimately bounded:

$$\|\mathbf{e}(t)\| \leq \max\Big(\sqrt{\frac{\lambda_P^+}{\lambda_P^-}} e^{-\frac{1-\lambda}{2\lambda_P^+}t}\|\mathbf{e}(0)\|, \frac{(\lambda_P^+)^{\frac{3}{2}}\delta}{\lambda\sqrt{\lambda_P^-}}\Big), \tag{28}$$

i. e. after a finite time T, the system error is in a ball around zero, with a radius proportional to the disturbance bound δ, which can be reduced thanks to the choice of the gains. Up to some dynamical feedback $\dot{\mathbf{w}} = a(\mathbf{x}, \mathbf{w}, \mathbf{v})$, $\mathbf{u} = b(\mathbf{x}, \mathbf{w}, \mathbf{v})$,

flatness induces that the mapping $\Theta : (\mathbf{x}, \mathbf{w}) \mapsto \mathbf{z}$ is a diffeomorphism, so its inverse preserves compacity, so the nonlinear system is practically stable in a compact lying in a ball of radius $r \leq \frac{(\lambda_P^+)^{\frac{3}{2}}\delta}{|\mathrm{d}\Theta|\lambda\sqrt{\lambda_P^-}}$. ■

IV. Numerical example

The proposed control strategy targets NCS or CPS (e. g., a fleet of robots) where data exchanges between entities are mandatory and should be encrypted for security reasons. The illustrative example concerns a single unicycle robot and it could be generalized to a robot fleet (the distributed control is not taken into account). We consider the trajectory tracking problem for a mobile robot described by the nonlinear system:

$$\left\{ \begin{array}{ll} \dot{x} & = v\cos\theta \\ \dot{y} & = v\sin\theta \\ \dot{\theta} & = \omega \end{array} \right. , \tag{29}$$

where $(x, y) \in \mathbb{R}^2$ are the coordinate of the center of mass of the mobile robot, $\theta \in \mathbb{S}^1$ is the orientation with respect to the x-axis, and $u = (v, \omega)^t$ are the linear and angular velocity controls. This system is well-known to be flat [16] with the flat outputs $(z_1, z_2) = (x, y)$, which we assume to be measured at all time. The dynamics of the flat outputs is given by the Brunovski form $\dot{\mathbf{z}} = A\mathbf{z} + B\mathbf{v}$, $z = C\mathbf{z}$, with $\mathbf{z} = (z_1, \dot{z}_1, z_2, \dot{z}_2)^t$, $\mathbf{v} = (v_1, v_2)^t$, and

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

According to equation (14), a classical Luenberger observer is designed on this Brunovski form to recover the states $(\dot{z}_1, \dot{z}_2)$ and thus the orientation θ of the mobile robot, which is necessary for implementing the stabilizing observer-based feedback $\mathbf{v}$ given by (15) and further (30). System (29) is flat, so we define a dynamic feedback (11) to recover the controls (v, ω) of the nonlinear system (29) from $\mathbf{v} = (v_1, v_2)$:

$$\begin{pmatrix} \dot{v} \\ \omega \end{pmatrix} = \frac{1}{v} \begin{pmatrix} v\cos\hat{\theta} & v\sin\hat{\theta} \\ -\sin\hat{\theta} & \cos\hat{\theta} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \tag{30}$$

where $\hat{\theta} = \arctan\left(\dot{\hat{z}}_4/\dot{\hat{z}}_2\right)$ is the value of θ reconstructed from the observer system.

We use the Python implementation of the CKKS scheme from [17], which is a binding with the state-of-the-art FHE library Microsoft SEAL [18]. We encrypt all the fixed data[1]: the matrices L and K, the reference trajectory $\mathbf{z}_{ref}$ and the reference control $\mathbf{v}_{ref}$. To homomorphically evaluate the dynamics of the observer and of the mobile robot using (14), (15), (29) and (30), we encrypt the flat outputs z, its derivative $\dot{z}$, the observed state $\hat{\mathbf{z}}$, and the values of $\cos\hat{\theta}$ and $\sin\hat{\theta}$.

For the numerical application, we set the reference trajectory to be the unit circle given by $\mathbf{z}_{ref}(t) = (\sin(t), \cos(t), -\cos(t), \sin(t))^t$, which provides the reference controls $\mathbf{v}_{ref}(t) = (-\sin(t), \cos(t))^t$ thanks to flatness. The eigenvalues for the observer are set to $(-4, -6, -4, -6)$ and for the feedback to $(-1, -3, -1, -3)$. The initialization of the mobile robot state vector is $(x, y, \theta)(0) = (0.5, 3.5, \pi/6)$, of the linear velocity is $v(0) = 1$, and of the observer is $\hat{\mathbf{z}} = 0$. The main parameters for the CKKS scheme are $N = 2^{14}$ for the ring dimension and $\Delta = 2^{40}$ for the scaling factor. Finally, the dynamics are integrated using Python routine *solve_ivp*. Figure 2 displays the numerical results for the each state (x, y, θ) of the nonlinear mobile robot. We also show the reference trajectory (in red) and, for comparison, a trajectory computed without encryption but with the same observer-based controller (in blue). We observe that the trajectory computed using the encrypted dynamics (in black) has a similar transient regime as the clear trajectory computed with the observer. Moreover, the encrypted trajectory converges towards the reference one at the same rate as the clear trajectory.

[1]Notice that these data can also to be in clear on the server if they are not considered sensible.

V. Concluding remarks

In this paper, we proposed a secure control architecture to address the security of CPS or NCS. The CKKS encryption algorithm allows to perform all the computations (integration of a continuous-time observer, state-feedback) on encrypted real numbers. The only limitation is to not being able to do operations other than addition and multiplication, and therefore not being able to manage nonlinear calculations. In order to circumvent this limitation, we have exploited the flatness property of nonlinear systems to get a linear equivalent system. We proved that the closed-loop system is practically stable despite the errors induced by encryption/decryption procedures (noise, quantization), assumed to be bounded. Simulations on a flat nonlinear system showed that an encrypted control strategy made possible to meet the control requirements in a secure way. The extension of this work to a fleet of robots is under investigation, handling real-time implementation concerns.

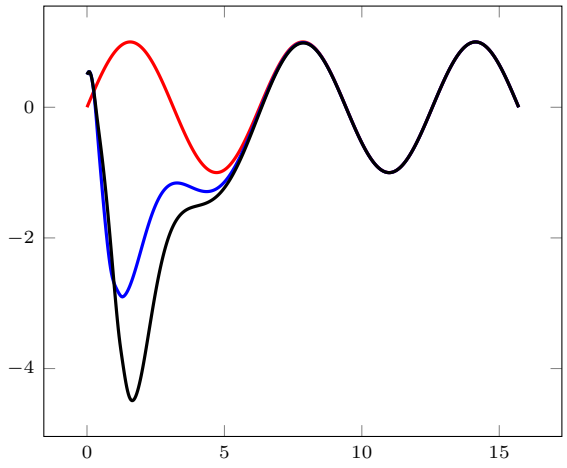
References

[1] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, *A secure control framework for resource-limited adversaries*, 2012.

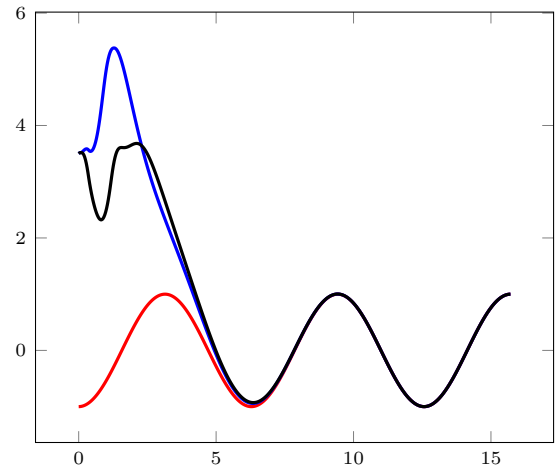
[2] M. N. Al-Mhiqani, R. Ahmad, W. Yassin, A. Hassan, Z. Z. Abidin, N. S. Ali, and K. H. Abdulkareem, "Cyber-security incidents: A review cases in cyber-physical systems," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 1, 2018.

[3] C. Gentry, "Fully homomorphic encryption using ideal lattices," *ACM Symposium on Theory of Computing*, 2009, pp. 169–178.

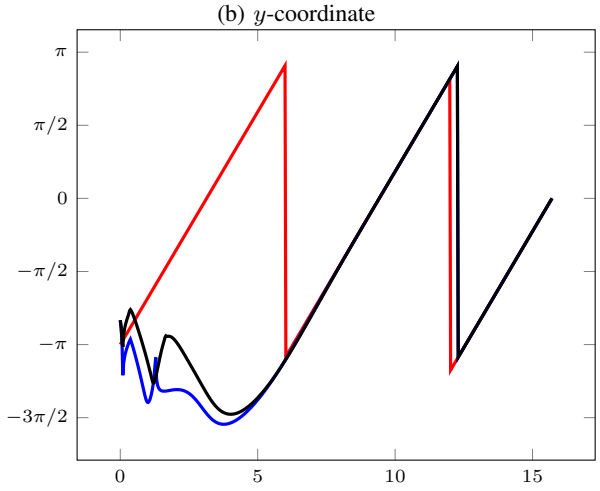
[4] K. Kogiso and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," *2015 54th Conference on Decision and Control*, 2015, pp. 6836–6843.



(a) x-coordinate



(b) y-coordinate



(c) θ-coordinate

Fig. 2: State (x, y, θ) of the mobile robot. In red is the reference trajectory, in blue the trajectory computed without encryption but with an observer-based controller, and in black is the trajectory computed using encrypted dynamics.

[5] N. Schlüter, P. Binfet, and M. Schulze Darup, "A brief survey on encrypted control: From the first to the second generation and beyond," *Annual Reviews in Control*, vol. 56, p. 100 913, 2023.

[6] H. Nguyen, B. Nguyen, H.-G. Lee, and H.-S. Ahn, *Encrypted observer-based control for linear continuous-time systems*, 2023.

[7] J. Tran, F. Farokhi, M. Cantoni, and I. Shames, "Implementing homomorphic encryption based secure feedback control," *Control Engineering Practice*, vol. 97, p. 104 350, 2020.

[8] M. Schulze Darup, A. Redder, and D. E. Quevedo, "Encrypted cooperative control based on structured feedback," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 37–42, 2019.

[9] C. Murguia, F. Farokhi, and I. Shames, "Secure and private implementation of dynamic controllers using semihomomorphic encryption," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3950–3957, 2020.

[10] Y. Lin, F. Farokhi, I. Shames, and D. Nešić, "Secure control of nonlinear systems using semi-homomorphic encryption," *IEEE Conference on Decision and Control*, 2018, pp. 5002–5007.

[11] F. Farokhi, I. Shames, and N. Batterham, "Secure and private control using semi-homomorphic encryption," *Control Engineering Practice*, vol. 67, pp. 13–20, 2017.

[12] M. Kishida, "Encrypted control system with quantiser," *IET Control Theory & Applications*, vol. 13, no. 1, pp. 146–151, 2019.

[13] J. Kim, H. Shim, and K. Han, "Dynamic controller that operates over homomorphically encrypted data for infinite time horizon," *IEEE Transactions on Automatic Control*, vol. 68, no. 2, pp. 660–672, 2023.

[14] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," *23rd International Conference on the Theory and Applications of Cryptology and Information Security*, Springer, 2017, pp. 409–437.

[15] I. Inc. "Introduction to the ckks encryption scheme." (2022), [Online]. Available: `https://www.inferati.com/blog/fhe-schemes-ckks#sec-evalmult`.

[16] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[17] A. Ibarrondo and A. Viand, "Pyfhel: Python for homomorphic encryption libraries," *Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2021, pp. 11–16.

[18] *Microsoft SEAL (release 4.1)*, `https://github.com/Microsoft/SEAL`, Microsoft Research, Redmond, WA., Jan. 2023.