Secure perception-driven control of mobile robots using chaotic encryption

X. Zhang, Z. Yuan, S. Xu, Y. Lu and M. Zhu

***Abstract*— This paper considers perception-driven control of a mobile robot for path tracking where perception is performed by a machine learning system. The robot is subject to passive attacks and evasion attacks on image transmission. To defeat the passive attacks, we adopt chaotic encryption technique to disguise pixels of plain images in real time, and construct a bank of fuzzy unknown input observers to decrypt the cipher pixels in parallel. We characterize the security level of the proposed chaotic cryptographic scheme. As for the path tracking, we derive a set of LMI conditions of the existence of a robust controller, which renders the output zeroing manifold attractive and invariant by using internal model technique, and also attenuates the effects of the evasion attacks and learning errors of the machine learning system by reducing $\mathcal{L}_2$ gain. Simulations are conducted in the CARLA simulator to demonstrate robust path tracking and secure image transmission.**

***Index Terms*— Robotic motion planning, perception, security**

I. Introduction

MOBILE robotic systems, e.g., unmanned aerial vehicles and self-driving cars, are becoming ubiquitous and have found a number of civilian and military applications [1]. Mobile robots integrate heterogeneous devices for embedded sensing, mobile computing and real-time control. These devices exchange information via on-board communication medium. Moreover, modern mobile robots adopt machine learning techniques to improve intelligence. However, the adoption of new technologies brings a wide spectrum of privacy and security issues. This paper specifically considers two classes of attacks. One is passive attacks on intra-robot communication, which can be launched to eavesdrop confidential information during data transmission. The other one is active attacks against machine learning systems at test time.

Literature review. In control systems, cryptography has been widely used to ensure data privacy during transmission. Existing works mainly focus on encrypting all the data in the control systems by homomorphic cryptosystems, such that all the operations in encrypted control systems are performed over encrypted data [3]. This common feature renders the sensitive data to be protected from eavesdroppers. However, homomorphic encryption could be slow and computationally expensive as key length increases [4], hence it is not suitable for high-dimensional image data encryption, especially when encryption needs to be done in real time. Since chaotic systems are extremely sensitive to initial states and system parameters, chaotic encryption methods can provide exceptionally good properties with regard to strong security and high speed [6]. However, chaotic encryption is also not integrated with control problems of dynamic systems.

Adversarial machine learning has been receiving increasing attention [23], [24]. In general, machine learning systems could be compromised during the training and test stages. First of all, training of a machine learning model requires an enormous amount of computational resources and data gathered by diverse sources, and adversaries can inject dirty samples into the training dataset to manipulate the model [7]. Secondly, evasion attacks at test time generate a set of elaborate samples to evade detection [8]. This set of results do not consider mitigation of the attacks on machine learning algorithms which are deployed on control systems.

There have been recent works which study attack-resilient estimation and control of robot systems. Availability, integrity, and confidentiality is the classic categorization of information security. Availability refers to that authorized users are able to access data whenever required. Denial-of-Service (DoS) attacks are commonly launched to compromise data availability and prevent legitimate users from accessing specific network resources [9]. Integrity refers to that data is not manipulated such that it keeps authentic, correct and reliable. Sensor attacks are typical examples to compromise data integrity and are studied in papers [10]. The above mentioned DoS attacks and sensor attacks are classified into active attacks and are well-studied. In contrast, passive attacks in robotic systems that can be launched to compromise data confidentiality [2] have not been sufficiently discussed yet.

Contributions. In this paper, we consider perception-driven control of a mobile robot for path tracking where perception is performed by a machine learning system. The robot is subject to passive attacks and evasion attacks on image transmission. Major contributions are listed as follows:

1) To defeat the passive attacks, we adopt chaotic encryption technique to disguise pixel values in the real-time plain images, and construct a bank of fuzzy unknown input observers to decrypt the cipher pixels in parallel.
2) We provide a sufficient condition of the equivalence between the proposed chaotic cryptographic scheme and conventional self-synchronizing stream cipher by using flatness.
3) We derive a set of linear matrix inequality (LMI) conditions of the existence of a robust controller, which renders the output zeroing manifold attractive and invariant by using internal model technique, and also attenuates the effects of the evasion attacks and learning errors of the machine learning system by reducing $\mathcal{L}_2$ gain.

The CARLA platform is used to conduct simulations on double integrator. The simulation results demonstrate robust path tracking and secure image transmission. Preliminary results of this paper were published in [12] where the system can track arbitrary differentiable paths under the assumptions on matching condition and observability. This paper relaxes these two assumptions, and instead assumes that the path is generated by a nonlinear exosystem. Moreover, this paper provides the theoretical and experimental results including attenuation to evasion attacks.

Notions and notations: Throughout the paper, we use $\mathbb{R}$ to represent the set of real numbers. The set of positive real numbers is denoted by $\mathbb{R}_+$. We use $\mathbb{R}^{m\times n}$ to denote the set of $m \times n$ real matrices. The set of m-dimensional symmetric positive definite matrices is denoted by $\mathbb{S}^m_+$. The complement of set $\mathcal{B}_1$ with respect to a set $\mathcal{B}_2$ is written by $\mathcal{B}_2/\mathcal{B}_1$. We denote $\mathcal{B}^n_r$ the ball centered at 0 with radius r in $\mathbb{R}^n$. A block diagonal matrix with submatrices $X_1, \ldots, X_p$ on

This work is supported by the awards NSF CNS 1505664 and ECCS 1846706.

X. Zhang, Z. Yuan, S. Xu and M. Zhu are with the School of Electrical Engineering and Computer Science, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: xxz313@psu.edu; zqy5086@psu.edu; spx5032@psu.edu; muz16@psu.edu).

Y. Lu is with the School of Computing and Communications, Lancaster University, Lancaster LA1 4WA, UK (e-mail: y.lu44@lancaster.ac.uk).

its main diagonal is denoted by $\mathrm{diag}\{X_1, \ldots, X_p\}$. For a matrix $\Gamma \in \mathbb{R}^{m\times n}$, Γ^{T} denotes its transpose, the hermitian operator $\mathscr{H}\{\cdot\}$ is defined as $\mathscr{H}\{\Gamma\} \triangleq \Gamma + \Gamma^{\mathrm{T}}$, the orthogonal complement matrix $\Gamma^{\perp}$ is defined as $\Gamma^{\perp}\Gamma = 0$ and $\Gamma^{\dagger} \triangleq (\Gamma^{\mathrm{T}}\Gamma)^{-1}\Gamma^{\mathrm{T}}$ is the left pseudo-inverse of Γ. We use $\sigma(\Gamma)$ to denote the spectrum of Γ. We use $\lambda_{\min}(\Gamma)$ and $\lambda_{\max}(\Gamma)$ to denote the minimal eigenvalue and maximal eigenvalue of matrix Γ, respectively. Moreover, we use the symbol $\star$ in a linear matrix inequality (LMI) to denote entries that follow from symmetry. Let A be an $n \times p$ matrix and B an $m \times q$ matrix. The Kronecker product of A and B is denoted by $A \otimes B$. We denote the ith column of A by $a_{\cdot,i} \triangleq [a_{1i}\ a_{2i}\ \ldots\ a_{ni}]^{\mathrm{T}}$, and the **vec** operator generates a column vector from a matrix A by stacking the column vectors of $A = [a_{\cdot,1}\, a_{\cdot,2} \ldots a_{\cdot,p}]$ below one another, i.e., $\mathbf{vec}(A) \triangleq \left[\begin{array}{ccc} a_{\cdot,1}^{\mathrm{T}} & \ldots & a_{\cdot,p}^{\mathrm{T}} \end{array}\right]^{\mathrm{T}}$.

For function $w(t) : [0, \infty) \to \mathbb{R}^m$, we denote $\|w_{[t_1,t_2]}\| \triangleq \sup_{t_1 \le t \le t_2} \|w(t)\|$. For function $f(x) : \mathbb{R}^n \to \mathbb{R}$, we denote by $\nabla f(x)$ its gradient at x.

A function $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ is of class $\mathcal{K}$ if it is continuous, $\gamma(0) = 0$, and strictly increasing; and is of class $\mathcal{K}_\infty$ if in addition it is unbounded. A function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ is of class $\mathcal{KL}$ if for each fixed $t \ge 0$, $\beta(s,t)$ is of class $\mathcal{K}$ and for each fixed $s \ge 0$, $\beta(s,t)$ decreases to zero as $t \to \infty$. Function composition is defined by $g \circ f(x) \triangleq g(f(x))$.

Consider a nonlinear system

$$\dot{x}(t) = f(x(t), w(t)), \quad y(t) = h(x(t)) \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is locally Lipschitz continuous in x and w with $f(0,0) = 0$, $h : \mathbb{R}^n \to \mathbb{R}^l$ is continuous with $h(0) = 0$ and $w(t)$ is a piecewise continuous, bounded function of t for all $t \ge 0$.

Definition 1: System (1) is input-to-state stable (ISS) from $w(t)$ to $x(t)$ if there exist a class $\mathcal{KL}$ function β and a class $\mathcal{K}$ function γ such that for any initial state $x(t_0)$ and any input $w(t)$, it holds that $\|x(t)\| \le \beta(\|x(t_0)\|, t - t_0) + \gamma\left(\|w_{[t_0,t]}\|\right)$ for all $t \ge t_0$.

The following flatness extends controllability from linear systems to nonlinear systems.

Definition 2: [21] System (1) is said to be flat if there exists output $y(t)$, referred to as flat output, such that state $x(t)$ and input $w(t)$ can be expressed as a function of the flat output $y(t)$ and a finite number of its derivatives.

II. Problem formulation

This section introduces secure perception-driven control of a mobile robot by applying chaotic encryption to communication of sensor readings.

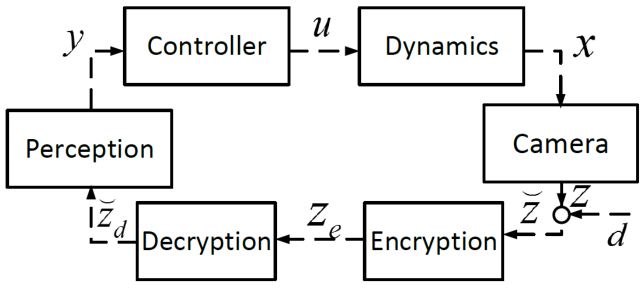


Fig. 1. Feedback loop of perception-driven control.

A. System model without encryption and evasion attacks

Consider the feedback loop in Fig. 1 where encryption and decryption, together with evasion attack $d(t)$ are excluded. The dynamic system of the robot is given by the following linear time-invariant system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) \tag{2}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the control signal and $y(t) \in \mathbb{R}^l$ is the output. The camera can generate state-dependent images, which is modeled by $z(t) = q(x(t))$ where $z(t) \in \mathbb{R}^{n_p}$ is the image vector and $n_p \gg n$. As [13], the image $z(t)$ passes through the perception unit, and the output is given as $y(t) = p(z(t))$ where $p : \mathbb{R}^{n_p} \to \mathbb{R}^l$ is the perception mapping. The output $y(t)$ is used to generate control command $u(t)$. Given a desired output matrix C, the perception mapping p is learned off-line from training data set $\{z(t), Cx(t)\}$ such that $p \circ q(x(t)) = Cx(t)$. Due to inherent learning errors, $p \circ q(x(t))$ could be different from $Cx(t)$. Then the output equation becomes $y(t) = Cx(t) + w(t)$ where $w(t) \triangleq p \circ q(x(t)) - Cx(t)$ represents the learning error of mapping p.

B. System model with encryption and evasion attacks

Now we consider the complete feedback loop in Fig. 1 where encryption, decryption, and evasion attacks are included. The 8-bit image data $z(t)$ can be tampered by evasion attacks $d(t) \in \mathbb{R}^{n_p}$. If the image is free of evasion attacks at time instant t then $d(t) = 0_{n_p}$, otherwise the evasion attacks can take any value from -255 to 255 to alter the pixels. We denote the corrupted image by $\breve{z}(t) \triangleq z(t) + d(t)$, which is transmitted through communication channels. In order to ensure confidentiality of the image, the camera encrypts the plain image using secret key Θ and sends the cipher image to the perception unit. Then the perception unit decrypts the encrypted image using Θ. The encryption mapping denoted by $\mathscr{E}_\Theta : \mathbb{R}^{n_p} \to \mathbb{R}^{n_e}$ is used to mask the corrupted plain image and the corresponding corrupted cipher image is represented as $z_e(t) \triangleq \mathscr{E}_\Theta(\breve{z}(t))$. The decryption mapping denoted by $\mathscr{D}_\Theta : \mathbb{R}^{n_e} \to \mathbb{R}^{n_p}$ is used to decrypt the cipher image and the decrypted image is represented by $\breve{z}_d(t) \triangleq \mathscr{D}_\Theta(z_e(t))$. The decrypted image $\breve{z}_d(t)$ passes through the perception unit and thus $y(t) = p(\breve{z}_d(t))$. By function composition, we get $y(t) = p \circ \mathscr{D}_\Theta \circ \mathscr{E}_\Theta \circ (q(x(t)) + d(t))$. Notice that $\mathscr{D}_\Theta \circ \mathscr{E}_\Theta(\breve{z}(t))$ may not be equal to $\breve{z}(t)$.

The overall dynamic system for Fig. 1 is given by

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + v(t) \tag{3}$$

where $v(t) \triangleq p \circ \mathscr{D}_\Theta \circ \mathscr{E}_\Theta \circ (q(x(t)) + d(t)) - p \circ q(x(t)) + w(t)$. Note that the difference between the first two terms represents the error caused by decryption and evasion attacks.

C. Control objective and assumptions

We consider a scenario where the robot dedicates to keeping track of a leading robot. We aim to design a secure perception-driven controller to achieve two goals:

(G1) image data $\breve{z}(t)$ is transmitted securely;

(G2) output $y(t)$ can keep track of the leading robot whose trajectory is generated by a nonlinear exosystem.

We impose a mild assumption on mappings p and q.

Assumption 1: Mappings p and q are continuous.

Neural networks are widely used for perception (see [14] and references therein). Since a neural network can be chosen by a composition of affine functions and continuous activation functions, e.g., sigmoid and ReLU, then p can be made continuous. By calibrating the camera, the pixel coordinates can be obtained from the world coordinate by linear matrix transformations, e.g., rotations and translations [15]. Then, the continuity of mapping q is satisfied.

The following assumption requires that the learning errors along the trajectory of system (2) are uniformly bounded. Remark 4 discusses how to design the perception unit p to satisfy the assumption.

Assumption 2: $\|w(t)\| \leq c_w$ for some $c_w > 0$ and all $t \geq 0$.

III. Main results

This section develops a secure perception-driven controller which includes two components. One is chaotic encryption which protects confidentiality of image data. The other is a robust path tracking controller which attenuates the error caused by decryption and evasion attacks.

A. Chaotic encryption

This section employs message-embedded chaotic encryption, and injects the plain image into a chaotic system. Many chaotic systems, e.g., Lorenz's system and Chua's circuit [6], can be written as a Takagi-Sugeno fuzzy system

$$\begin{aligned} \dot{x}_e(t) &= \textstyle\sum_{i=1}^N \mu_i(\xi(t))(A_{e,i}x_e(t)) \\ z_e(t) &= \textstyle\sum_{i=1}^N \mu_i(\xi(t))(C_{e,i}x_e(t)) \end{aligned} \tag{4}$$

where $x_e(t) \in \mathbb{R}^{n_s}$, $z_e(t) \in \mathbb{R}^{n_e}$ are the state and output vectors, respectively; N is the number of subsystems, $A_{e,i}$ and $C_{e,i}$ are matrices with appropriate dimensions. The weighting functions $\mu_i(\xi)$ depend on parameter vector ξ and satisfy the convex sum property $\sum_{i=1}^N \mu_i(\xi) = 1$ and $0 \leq \mu_i(\xi) \leq 1$. Note that ξ is usually a function of measurable state variables. As [20] defined, system (4) exhibiting chaos is commonly referred to as a transmitter.

The camera maintains a message-embedded cryptosystem for each pixel $\check{z}_j(t) \in \mathbb{R}$, and the cryptosystems are in the same form and executed in parallel. To simplify notations, we remove the subscript j of each pixel $\check{z}_j(t)$, and directly use $\check{z}(t)$ to represent a pixel in the remaining of this section. In particular, each pixel $\check{z}(t) \in \mathbb{R}$ of the corrupted plain image is injected into the transmitter (4) with zero feedthrough matrix, which becomes the following cryptosystem

$$\begin{aligned} \dot{x}_e(t) &= \textstyle\sum_{i=1}^N \mu_i(\xi(t))\left(A_{e,i}x_e(t)\right) + B_e\check{z}(t) \\ z_e(t) &= C_e x_e(t). \end{aligned} \tag{5}$$

In above, the pixel $\check{z}(t)$ acts as an unknown input of the transmitter whose state is $x_e(t) \in \mathbb{R}^{n_s}$ and output is $z_e(t) \in \mathbb{R}^{n_e}$. We represent B_e and C_e as the input matrix and output matrix, respectively. In cryptography, $x_e(t)$ can be considered as keystream. For each cryptosystem (5), the initial state $x_e(0)$ as well as the matrices $A_{e,i}, B_e, C_e$ can be considered as part of the secret key Θ. The perception unit aims to use $z_e(t)$ and $\dot{z}_e(t)$ to recover the corrupted plain pixel $\check{z}(t)$. This can be achieved via an unknown input observer (UIO).

In this paper, we pick a non-singular matrix C_e such that $\text{rank}(C_eB_e) = \text{rank}B_e$ in the design of the cryptosystem (5). Then we decompose cryptosystem (5) into two subsystems: one is free of the unknown input $\check{z}(t)$, and the other is dependent on it. Matrices $T_e \in \mathbb{R}^{n_s \times n_s}$ and $U_e \in \mathbb{R}^{n_e \times n_e}$ are defined as

$$T_e \triangleq \begin{bmatrix} B_e^{\perp} \\ (C_eB_e)^{\dagger}C_e \end{bmatrix}, \quad U_e \triangleq \begin{bmatrix} (C_eB_e)^{\perp} \\ (C_eB_e)^{\dagger} \end{bmatrix}. \tag{6}$$

Note that T_e is nonsingular since $T_e^{-1}T_e = I_{n_s}$ where $T_e^{-1} \triangleq \begin{bmatrix} \tilde{T}_e & B_e \end{bmatrix}$ and $\tilde{T}_e \triangleq \left[I_{n_s} - B_e(C_eB_e)^{\dagger}C_e\right](B_e^{\perp})^{\dagger}$. With the state transformation $\bar{x}_e \triangleq T_e x_e$ and output transformation $\bar{z}_e \triangleq U_e z_e$, cryptosystem (5) is partitioned into a new form

$$\begin{aligned} \dot{\bar{x}}_{e,1}(t) &= \textstyle\sum_{i=1}^N \mu_i(\xi(t))\left(A_{e,i}^1\bar{x}_{e,1}(t) + A_{e,i}^2\bar{x}_{e,2}(t)\right) \\ \dot{\bar{x}}_{e,2}(t) &= \textstyle\sum_{i=1}^N \mu_i(\xi(t))\Big(A_{e,i}^3\bar{x}_{e,1}(t) \\ &\qquad\qquad + A_{e,i}^4\bar{x}_{e,2}(t) + \check{z}(t)\Big) \\ \bar{z}_{e,1}(t) &= \bar{C}_e\bar{x}_{e,1}(t), \bar{z}_{e,2}(t) = \bar{x}_{e,2}(t) \end{aligned}$$

where $\bar{x}_e(t) \triangleq [\bar{x}_{e,1}(t)^{\mathrm{T}}, \bar{x}_{e,2}(t)^{\mathrm{T}}]^{\mathrm{T}}$, $\bar{z}_e(t) \triangleq [\bar{z}_{e,1}(t)^{\mathrm{T}}, \bar{z}_{e,2}(t)^{\mathrm{T}}]^{\mathrm{T}}$, $\bar{x}_{e,1}(t) \in \mathbb{R}^{n_s-1}, \bar{x}_{e,2}(t) \in \mathbb{R}$, $\bar{z}_{e,1}(t) \in \mathbb{R}^{n_e-1}$, $\bar{z}_{e,2}(t) \in \mathbb{R}$, and

$$\begin{aligned} &A_{e,i}^1 \triangleq B_e^{\perp}A_{e,i}\tilde{T}_e, \; A_{e,i}^2 \triangleq B_e^{\perp}A_{e,i}B_e, \; \bar{C}_e \triangleq (C_eB_e)^{\perp}C_e\tilde{T}_e \\ &A_{e,i}^3 \triangleq (C_eB_e)^{\dagger}C_eA_{e,i}\tilde{T}_e, \quad A_{e,i}^4 \triangleq (C_eB_e)^{\dagger}C_eA_{e,i}B_e. \end{aligned}$$

It indicates that $\bar{x}_{e,2}(t)$ can be directly recovered from $\bar{z}_{e,2}(t)$, i.e., $\bar{x}_{e,2}(t) = \bar{z}_{e,2}(t)$. We use the following unknown-input-free subsystem to reconstruct $\bar{x}_{e,1}(t)$

$$\begin{aligned} \dot{\bar{x}}_{e,1}(t) &= \textstyle\sum_{i=1}^N \mu_i(\xi(t))\left(A_{e,i}^1\bar{x}_{e,1}(t) + A_{e,i}^2\bar{x}_{e,2}(t)\right) \\ \bar{z}_{e,1}(t) &= \bar{C}_e\bar{x}_{e,1}(t), \end{aligned}$$

and then the state $x_e(t)$ can be recovered as follows:

$$x_e(t) = T_e^{-1}\begin{bmatrix} \bar{x}_{e,1}(t) \\ \bar{x}_{e,2}(t) \end{bmatrix} = T_e^{-1}\begin{bmatrix} \bar{x}_{e,1}(t) \\ (C_eB_e)^{\dagger}z_e(t) \end{bmatrix}. \tag{7}$$

The perception unit uses the following state observer to recover $x_e(t)$

$$\begin{aligned} \dot{\hat{\bar{x}}}_{e,1}(t) &= \textstyle\sum_{i=1}^N \mu_i(\xi(t))\Big(A_{e,i}^1\hat{\bar{x}}_{e,1}(t) \\ &\qquad + A_{e,i}^2\bar{z}_{e,2}(t) + L_{e,i}(\bar{C}_e\hat{\bar{x}}_{e,1}(t) - \bar{z}_{e,1}(t))\Big) \\ \hat{x}_e(t) &= T_e^{-1}\begin{bmatrix} \hat{\bar{x}}_{e,1}(t) \\ (C_eB_e)^{\dagger}z_e(t) \end{bmatrix} \end{aligned} \tag{8}$$

where $\hat{\bar{x}}_{e,1}(t)$ is the estimate of $\bar{x}_{e,1}(t)$, and $L_{e,i}$, $i = 1, \ldots, N$, are the observer gains. Differentiating the output equation of cryptosystem (5) with regard to t and replacing $x_e(t)$ with $\hat{x}_e(t)$, the decrypted pixel is given as

$$\check{z}_d(t) = \textstyle\sum_{i=1}^N \mu_i(\xi(t))(C_eB_e)^{\dagger}\left(\dot{z}_e(t) - C_eA_{e,i}\hat{x}_e(t)\right), \tag{9}$$

and the decrypted image consists of $\check{z}_d(t)$. In theory, $\dot{z}_e(t)$ is required for the above derivation of the decrypted pixel $\check{z}_d(t)$. We define the state estimation error as $\tilde{\bar{x}}_{e,1}(t) \triangleq \bar{x}_{e,1}(t) - \hat{\bar{x}}_{e,1}(t)$, then by the obtained state observer (8), we derive the error dynamics

$$\dot{\tilde{\bar{x}}}_{e,1}(t) = \textstyle\sum_{i=1}^N \mu_i(\xi(t))\left((A_{e,i}^1 - L_{e,i}\bar{C}_e)\tilde{\bar{x}}_{e,1}(t)\right). \tag{10}$$

The following lemma employs a common Lyapunov function to derive a sufficient condition for exponential convergence of estimation errors, which guarantees the synchronization between the state $x_e(t)$ of the cryptosystem (5) and the state $\hat{x}_e(t)$ of the receiver (8).

Lemma 1: If there exist matrices $P_e \in \mathbb{S}_+^{n_s-1}$, $Q_{e,i} \in \mathbb{R}^{(n_s-1)\times(n_e-1)}$, $\forall i \in \{1, \ldots, N\}$ and scalar $\gamma_e \in \mathbb{R}_+$, to satisfy the following LMI conditions

$$\begin{aligned} &(A_{e,i}^1)^{\mathrm{T}}P_e + P_eA_{e,i}^1 - (\bar{C}_e)^{\mathrm{T}}Q_{e,i}^{\mathrm{T}} - Q_{e,i}\bar{C}_e + \gamma_e I < 0, \\ &\forall i = 1, \ldots, N, \end{aligned} \tag{11}$$

the error dynamics (10) is globally exponentially stable with observer gain matrices $L_{e,i} = P_e^{-1}Q_{e,i}$, and $\|\check{z}(t) - \check{z}_d(t)\|$ diminishes exponentially.

The proof of Lemma 1 is present in Appendix VI.A of [27]. Next, we discuss the security level of our chaotic encryption method. The following lemma theoretically shows the equivalence between the message-embedded cryptosystem and the conventional self-synchronizing stream cipher.

Lemma 2: The message-embedded cryptosystem (5) is equivalent to a conventional self-synchronizing stream cipher.

The proof of Lemma 2 is given in Appendix VI.B of [27].

Remark 1: By Lemma 2 and Proposition 1 in [20], our chaotic encryption is able to provide the same level of security as a conventional self-synchronizing stream cipher.

Remark 2: Notice that $\dot{z}_e(t)$ is used to theoretically guarantee the recovery of the unknown input, and it is a standard and necessary requirement for the continuous-time UIO technique (see [11] and references therein). Notice that $\dot{z}_e(t)$ is well-defined only when $z_e(t)$ is real-valued. However, in this paper, $z_e(t)$ is the value of a pixel and an integer. In practice, we replace $\dot{z}_e(t)$ by $\frac{z_e(t)-z_e(t-\tau)}{\tau}$ where τ is the sampling period.

B. Robust tracking controller

The robot aims to follow the leading robot whose trajectory is generated by the following nonlinear measurable exosystem, which includes the Van der Pol oscillator,

$$\begin{aligned}&\dot{x}_r(t) = A_r x_r(t) + \textstyle\sum_{i=1}^K E_i x_r(t) a_i(x_r(t))\\ &r(t) = C_r x_r(t)\end{aligned} \tag{12}$$

where $x_r(t) \in \mathbb{R}^{n_r}$ is the state, $a_i : \mathbb{R}^{n_r} \to \mathbb{R}$ is continuously differentiable with $a_i(0) = 0$ and A_r, C_r, E_i, $i = 1, \ldots, K$, are matrices with appropriate dimensions. Combining (12) with the controlled system (3) as well as $e(t) = y(t) - r(t)$ renders the following augmented system

$$\begin{aligned}&\dot{x}(t) = Ax(t) + Bu(t)\\ &\dot{x}_r(t) = A_r x_r(t) + \textstyle\sum_{i=1}^K E_i x_r(t) a_i(x_r(t))\\ &e(t) = Cx(t) - C_r x_r(t) + v(t).\end{aligned} \tag{13}$$

We decompose $v(t)$ in such a way that $v(t) = \Delta(t) + w(t)$ where $\Delta(t) \triangleq p(\check{z}_d(t)) - p(z(t))$. The following lemma gives an upper bound of $\Delta(t)$.

Lemma 3: Let the assumptions of Lemma 1 and Assumption 1 hold. Then there exists a constant $c_\Delta > 0$ such that $\|\Delta(t)\| \le c_\Delta$ for all $t \ge 0$.

The proof of Lemma 3 is given in Appendix VI.C of [27].

Assume $v(t) = 0$. We would like to determine sufficiently smooth functions $\boldsymbol{x}(x_r)$ and $\boldsymbol{u}(x_r)$ with $\boldsymbol{x}(0) = 0$ and $\boldsymbol{u}(0) = 0$ such that the tracking error is constantly zero when the state of system (13) is restricted to the output zeroing manifold $\mathcal{M}_1 \triangleq \{(x, x_r) \in \mathbb{R}^n \times \mathbb{R}^{n_r} | x = \boldsymbol{x}(x_r)\}$ under controller $u = \boldsymbol{u}(x_r)$. Substituting $x = \boldsymbol{x}(x_r)$ and $u = \boldsymbol{u}(x_r)$ into dynamics (13) renders the following regulator equation

$$\begin{aligned}A\boldsymbol{x}(x_r) + B\boldsymbol{u}(x_r) &= \frac{\partial \boldsymbol{x}(x_r)}{\partial x_r}\left(A_r x_r + \sum_{i=1}^K E_i x_r a_i(x_r)\right)\\ C\boldsymbol{x}(x_r) - C_r x_r &= 0.\end{aligned} \tag{14}$$

Assumption 3: Solutions $\boldsymbol{x}(x_r)$ and $\boldsymbol{u}(x_r)$ of regulator equation (14) exist and $\boldsymbol{u}(x_r)$ is polynomial in x_r.

The existence of solution $x = \boldsymbol{x}(x_r)$ and $u = \boldsymbol{u}(x_r)$ only ensures the tracking error $e = 0$. To guarantee the boundedness of all trajectories of the closed-loop system associated with system (13) and controller $u(t)$, the assumption that $\boldsymbol{u}(x_r)$ is polynomial in x_r is required [18]. Since it is hard to solve regulator equation (14), we first design a steady-state generator to reproduce $\boldsymbol{u}(x_r)$. Second, we design an internal model to reconstruct $\boldsymbol{u}(x_r)$.

Letting $\boldsymbol{x}(x_r) = \Pi x_r$ with a matrix $\Pi \in \mathbb{R}^{n_r \times n_r}$. Since the solution $\boldsymbol{u}(x_r(t))$ is a polynomial in $x_r(t)$, there exists a set of matrices $\Lambda_1, \Lambda_2, \ldots, \Lambda_r \in \mathbb{R}^{m\times m}$ for some positive integer r, such that

$$L^r_{A_r x_r}\boldsymbol{u} = \Lambda_1 \boldsymbol{u} + \Lambda_2 L_{A_r x_r}\boldsymbol{u} + \ldots + \Lambda_r L^{r-1}_{A_r x_r}\boldsymbol{u},$$

where $L_{A_r x_r}\boldsymbol{u} \triangleq \frac{\partial \boldsymbol{u}}{\partial x_r} A_r x_r$, and $L^j_{A_r x_r}\boldsymbol{u} \triangleq \frac{\partial L^{j-1}_{A_r x_r}\boldsymbol{u}}{\partial x_r} A_r x_r$, $j = 2, 3, \ldots, r$. Denote

$$\theta(x_r) \triangleq [\boldsymbol{u}(x_r)^{\mathrm{T}}, (L_{A_r x_r}\boldsymbol{u}(x_r))^{\mathrm{T}}, \ldots, (L^{r-1}_{A_r x_r}\boldsymbol{u}(x_r))^{\mathrm{T}}]^{\mathrm{T}}.$$

There exist $\Phi \triangleq \begin{bmatrix} 0_{m(r-1)\times m} & I_{m(r-1)} \\ \Lambda_1 & [\Lambda_2 \cdots \Lambda_r] \end{bmatrix}$ and $\Psi \triangleq [I_m\ 0_m\ \ldots\ 0_m]$ such that

$$\frac{\partial \theta(x_r)}{\partial x_r} A_r x_r = \Phi\theta(x_r), \quad \boldsymbol{u}(x_r) = \Psi\theta(x_r). \tag{15}$$

Assumption 4: There exists some matrix Φ_i satisfying

$$\frac{\partial \theta(x_r)}{\partial x_r} E_i x_r = \Phi_i \theta(x_r),\ i = 1, \ldots, K. \tag{16}$$

As [18] shows, Assumption 4 may hold in many cases, e.g., Van der Pol oscillator. Assumptions 3 and 4 imply that system (13) has a steady-state generator with output $\boldsymbol{u}$. Let $\hat{\theta}(x_r(t)) \triangleq \Omega\theta(x_r(t))$ with any non-singular matrix Ω, and take Lie derivative on both sides along system (12). By using (16), the steady-state generator is constructed as

$$\begin{aligned}\dot{\hat{\theta}}(x_r(t)) &= \Omega(\Phi + \phi(x_r(t)))\Omega^{-1}\hat{\theta}(x_r(t))\\ \boldsymbol{u}(x_r(t)) &= \beta(\hat{\theta}(x_r(t))) = \Psi\Omega^{-1}\hat{\theta}(x_r(t))\end{aligned} \tag{17}$$

where $\phi(x_r(t)) \triangleq \sum_{i=1}^K \Phi_i a_i(x_r(t))$.

We design a nonlinear internal model candidate as follows. We pick any controllable pair (F, G) with $F \in \mathbb{R}^{mr\times mr}$ being Hurwitz and $G \in \mathbb{R}^{mr\times m}$. By [17], there exists a nonsingular matrix $\Omega \in \mathbb{R}^{mr\times mr}$ as the unique solution of the following Sylvester equation

$$\Omega\Phi - F\Omega = G\Psi. \tag{18}$$

By (15) and (18), the steady-state generator (17) is written as

$$\dot{\hat{\theta}}(x_r(t)) = F\hat{\theta}(x_r(t)) + \Omega\phi(x_r(t))\Omega^{-1}\hat{\theta}(x_r(t)) + G\boldsymbol{u}(x_r(t)).$$

Then an internal model candidate is constructed as

$$\dot{\eta}(t) = F\eta(t) + \Omega\phi(x_r(t))\Omega^{-1}\eta(t) + Gu(t). \tag{19}$$

Applying the following state and input transformations

$$\begin{aligned}&\tilde{x}(t) \triangleq x(t) - \Pi x_r(t), \quad \tilde{\eta}(t) \triangleq \eta(t) - \hat{\theta}(x_r(t)),\\ &\tilde{u}(t) \triangleq u(t) - \beta(\eta(t))\end{aligned}$$

yields an error dynamics

$$\begin{aligned}\dot{\tilde{x}}(t) &= A\tilde{x}(t) + B\Psi\Omega^{-1}\tilde{\eta}(t) + B\tilde{u}(t)\\ &\quad + B\Psi\theta(x_r(t)) - \Pi a(x_r(t)) + A\Pi x_r(t)\\ \dot{\tilde{\eta}}(t) &= (F + G\Psi\Omega^{-1} + \Omega\phi(x_r(t))\Omega^{-1})\tilde{\eta}(t) + G\tilde{u}(t)\\ e(t) &= C\tilde{x}(t) + v(t).\end{aligned}$$

We let $x_r(t) = 0$, and the resulting system is simplified as

$$\begin{aligned}\dot{\tilde{x}}(t) &= A\tilde{x}(t) + B\Psi\Omega^{-1}\tilde{\eta}(t) + B\tilde{u}(t)\\ \dot{\tilde{\eta}}(t) &= (F + G\Psi\Omega^{-1})\tilde{\eta}(t) + G\tilde{u}(t)\\ e(t) &= C\tilde{x}(t) + v(t).\end{aligned} \tag{20}$$

A dynamic error compensator is chosen as

$$\dot{x}_p(t) = A_p x_p(t) + B_p e(t), \quad \tilde{u}(t) = C_p x_p(t). \tag{21}$$

where $x_p(t) \in \mathbb{R}^{n_c}$ is state and A_p, B_p, C_p are controller parameters to be determined. Substituting controller (21) into (20) renders a closed-loop system

$$\begin{aligned}\dot{\tilde{x}}_{cl}(t) &= \tilde{A}_{cl}\tilde{x}_{cl}(t) + \tilde{B}_{cl}\Delta(t) + \tilde{B}_{cl}w(t)\\ e(t) &= \tilde{C}_{cl}\tilde{x}_{cl}(t) + \Delta(t) + w(t)\end{aligned} \tag{22}$$

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2023.3312581

where $\tilde{x}_{cl}(t) \triangleq [\tilde{x}(t)^{\mathrm{T}} \ \tilde{\eta}(t)^{\mathrm{T}} \ x_p(t)^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{n+mr+n_c}$, $\tilde{A}_{cl} \triangleq \begin{bmatrix} A' & B'C_p \\ B_pC' & A_p \end{bmatrix}$, $\tilde{B}_{cl} \triangleq \begin{bmatrix} 0 \\ B_p \end{bmatrix}$, $\tilde{C}_{cl} \triangleq [C' \quad 0]$ and $A' \triangleq \begin{bmatrix} A & B\Psi\Omega^{-1} \\ 0 & F+G\Psi\Omega^{-1} \end{bmatrix}$, $B' \triangleq \begin{bmatrix} B \\ G \end{bmatrix}$, and $C' \triangleq \begin{bmatrix} C & 0 \end{bmatrix}$. The following lemma shows the convergence to a neighborhood of the output zeroing manifold $\mathcal{M}_2 \triangleq \{(x, \eta, x_r) \in \mathbb{R}^n \times \mathbb{R}^{mr} \times \mathbb{R}^{n_r} | x = \Pi x_r, \eta = \hat{\theta}(x_r)\}$.

Lemma 4: If there exist $P \in \mathbb{S}_+^{n+mr+n_c}$ and $\gamma_p \in \mathbb{R}_+$ such that

$$\begin{bmatrix} \mathscr{H}\{P\tilde{A}_{cl}\} + P & P\tilde{B}_{cl} & \tilde{C}_{cl}^{\mathrm{T}} \\ \tilde{B}_{cl}^{\mathrm{T}}P & -(\gamma_p - 1)I_l & I_l \\ \tilde{C}_{cl} & I_l & -(\gamma_p - 1)I_l \end{bmatrix} < 0, \tag{23}$$

then system (22) is ISS from $w(t)$ and $\Delta(t)$ to $\tilde{x}_{cl}(t)$.

The proof of Lemma 4 is provided in Appendix VI.D of [27].

Attaching the internal model (19) to the given system (3) renders the following augmented system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ \dot{\eta}(t) &= F\eta(t) + \Omega\phi(x_r(t))\Omega^{-1}\eta(t) + Gu(t) \\ e(t) &= Cx(t) - C_rx_r(t) + v(t). \end{aligned} \tag{24}$$

Then, the path tracking problem is converted into a stabilization problem of the augmented system (24). By (17) and (21), an output-feedback controller is given by

$$\dot{x}_p(t) = A_px_p(t) + B_pe(t), u(t) = C_px_p(t) + \Psi\Omega^{-1}\eta(t) \tag{25}$$

and substituting it into the augmented system (24) yields the following closed-loop system

$$\begin{aligned} \dot{x}_{cl}(t) &= A_{cl}(x_r(t))x_{cl}(t) + B_{cl}(x_r(t), v(t)) \\ e(t) &= C_{cl}x_{cl}(t) - C_rx_r(t) + v(t) \end{aligned} \tag{26}$$

where $x_{cl}(t) \triangleq [x(t)^{\mathrm{T}} \ \eta(t)^{\mathrm{T}} \ x_p(t)^{\mathrm{T}}]^{\mathrm{T}}$, $A_{cl}(x_r) \triangleq \begin{bmatrix} A & B\Psi\Omega^{-1} & BC_p \\ 0 & F + G\Psi\Omega^{-1} + \Omega\phi(x_r)\Omega^{-1} & GC_p \\ B_pC & 0 & A_p \end{bmatrix}$, and $B_{cl}(x_r, v) \triangleq \begin{bmatrix} 0 \\ 0 \\ -B_pC_rx_r + B_pv \end{bmatrix}$.

Considering a ball $\mathcal{B}_{\hat{\nu}}^n$ with the radius $\hat{\nu}$, the following lemma is used in the proof of our main results.

Lemma 5: If $f(x) : \mathbb{R}^n \to \mathbb{R}$ is continuous on $\mathbb{R}^n$, then $g(\hat{\nu}) \triangleq \max_{x \in \mathcal{B}_{\hat{\nu}}^n} f(x)$ is continuous on $\mathbb{R}_+$.

The proof of Lemma 5 is given in Appendix VI.E of [27]. We pick any $\nu > 0$ such that $\|x_r(t)\| \leq \nu$ for $t \geq 0$. The following theorem shows a sufficient condition of the local stability of closed-loop system (26).

Theorem 1: Let the assumptions of Lemma 3 and Assumption 2 hold. Suppose that there exist $R \in \mathbb{S}_+^{n+mr}, S \in \mathbb{S}_+^{n+mr}$, and matrices $\hat{A}_p \in \mathbb{R}^{n_c \times n_c}$, $\hat{B}_p \in \mathbb{R}^{n_c \times l}$, $\hat{C}_p \in \mathbb{R}^{m \times n_c}$, and scalar $\gamma_p \in \mathbb{R}_+$ such that the following LMIs are feasible

$$\left[\begin{array}{cc:c:c} \mathscr{H}\{A'R + B'\hat{C}_p\} + R & \star & \star & \star \\ \hat{A}_p + A'^{\mathrm{T}} + I_{n+mr} & \mathscr{H}\{SA' + \hat{B}_pC'\} + S & \star & \star \\ \hdashline 0 & \hat{B}_p^{\mathrm{T}} & -(\gamma_p - 1)I_l & \star \\ \hdashline C'R & C' & I_l & -(\gamma_p - 1)I_l \end{array}\right] < 0, \tag{27}$$

$$\begin{bmatrix} R & I_{n+mr} \\ I_{n+mr} & S \end{bmatrix} > 0. \tag{28}$$

Then the following properties hold.

(P1) The scalar γ_p and the positive-definite matrix $P = \begin{bmatrix} S & Y \\ Y^{\mathrm{T}} & I_{n_c} \end{bmatrix}$ satisfy (23), and the parameters A_p, B_p, C_p, in controller (25) can be computed by

$$\begin{bmatrix} A_p & B_p \\ C_p & 0 \end{bmatrix} = \begin{bmatrix} Y & SB' \\ 0 & I_l \end{bmatrix}^{-1} \times \left(\begin{bmatrix} \hat{A}_p & \hat{B}_p \\ \hat{C}_p & 0 \end{bmatrix} - \begin{bmatrix} SA'R & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} M^{\mathrm{T}} & 0 \\ C'R & I_l \end{bmatrix}^{-1} \tag{29}$$

where matrices $M, Y \in \mathbb{R}^{n_c \times n_c}$ have full rank and satisfy $YM^{\mathrm{T}} = I_{n_c} - RS$.

(P2) Choose any $R_0 > 0$ and $\delta \in (0, 1)$. Let $R_s \triangleq \sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}R_0 + \frac{2\|P\|\|B_p\|\|C_r\|}{\lambda_{\min}(P)\delta}(\nu + \|C_r\|^{-1}(c_w + c_\Delta))$. For a sufficiently small ν, if $x_{cl}(0) \in \mathcal{B}_{R_0}^{n+mr+n_c}$, system (26) satisfies the following properties:

(P2.1) ISS from $w(t)$, $\Delta(t)$ and $x_r(t)$ to $x_{cl}(t)$;
(P2.2) $x_{cl}(t) \in \mathcal{B}_{R_s}^{n+mr+n_c}$ for all $t \geq 0$;
(P2.3) $\lim_{T \to \infty} \frac{\int_0^T \|e(t)\|^2 dt}{\int_0^T \|\Delta(t) + w(t)\|^2 dt} < \gamma_p^2$.

The proof of Theorem 1 is given in Appendix VI.F of [27].

Remark 3: One can check the feasibility of the LMI conditions (27)-(28) using *Robust Control Toolbox* in Matlab [26].

Remark 4: Theorem 1 provides a guideline to design a neural network for the perception unit p to satisfy Assumption 2. First, choose any bound $c_w > 0$ for learning error w. Second, choose a pair of R_0 and δ and compute R_s. Third, by universal approximation property (see Theorem 2 in [22]), for any compact set, a standard multilayer feedforward network with a sufficiently large number of hidden-layer neurons is able to approximate any continuous function to any degree of accuracy if the activation functions are continuous, bounded and nonconstant. Hence we can choose a neural network with sigmoid activation functions for the perception unit p, and increase the number of hidden-layer neurons such that $\|w(x)\| \leq c_w$ for all $x \in \mathcal{B}_{R_s}^n$. By Theorem 1, if $x_{cl}(0) \in \mathcal{B}_{R_0}^{n+mr+n_c}$, then $x_{cl}(t) \in \mathcal{B}_{R_s}^{n+mr+n_c}$ and $\|w(t)\| \leq c_w$ for all $t \geq 0$, i.e., Assumption 2 holds.

IV. Simulation

This section provides a simulation by using double integrators in the CARLA simulator [16]. The computer used in the simulation has Core $i7 - 3632$ QM CPU with 2.20 GHz and 15.5 GiB Memory.

A. System model

Consider a robot moving in a 2-D plane. The dynamics of the double integrator for horizontal and vertical directions are compactly given by

$$\begin{aligned} \dot{x}(t) &= \left[\begin{array}{cc:cc} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right] x(t) + \left[\begin{array}{c:c} 0 & 0 \\ 1 & 0 \\ \hdashline 0 & 0 \\ 0 & 1 \end{array}\right] u(t), \\ y(t) &= \left[\begin{array}{cc:cc} 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 \end{array}\right] x(t) \end{aligned}$$

where state $x(t)$ includes positions $x_1(t)$, $x_3(t)$ and velocities $x_2(t)$, $x_4(t)$, output $y(t)$ includes positions, and control $u(t)$ is acceleration. The camera model is $z(t) = q(x(t))$ where $z(t)$ is the plain image. We learn the perception mapping p by a convolutional neural network (CNN), which has been widely applied to image detection and recognition.

B. Controller and UIO design

This subsection considers the case where the trajectory of the leader robot is generated by a Van der Pol oscillator

$$\begin{bmatrix}\dot{x}_{r1}(t)\\ \dot{x}_{r2}(t)\end{bmatrix} = \begin{bmatrix} x_{r2}(t) \\ -x_{r1}(t)+\left(1-x_{r1}(t)^2\right)x_{r2}(t)\end{bmatrix}$$
$$= A_r x_r(t) + E_1 x_r(t) a_1(x_r(t))$$

with $A_r = \begin{bmatrix} 0 & 1 \\ -1 & 1\end{bmatrix}$, $E_1 = \begin{bmatrix} 0 & 0 \\ 0 & -1\end{bmatrix}$ and $a_1(x_r(t)) = x_{r1}(t)^2$. We let two double integrators track x_{r1} and x_{r2}, respectively. For brevity, we only give the controller design details when the double integrator in the horizontal direction follows x_{r1}. In this case, system matrices are given by $A = \begin{bmatrix} 0 & 1 \\ 0 & 0\end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1\end{bmatrix}$, $C = \begin{bmatrix} 1 & 0\end{bmatrix}$ and $C_r = \begin{bmatrix} 1 & 0\end{bmatrix}$. The solution of regulator equation (14) is given as $\boldsymbol{x}(x_r) = \begin{bmatrix} x_{r1} & x_{r2}\end{bmatrix}^{\mathrm{T}}, \boldsymbol{u}(x_r) = -x_{r1} + \left(1-x_{r1}^2\right)x_{r2}$. Then Assumption 3 is satisfied. As [18], to simplify the controller design, we let $\boldsymbol{u}_c(x_r) = \left(1-x_{r1}^2\right)x_{r2}$ and $\hat{\boldsymbol{u}}(x_r) = -x_{r1}$ such that $\boldsymbol{u}(x_r) = \boldsymbol{u}_c(x_r) + \hat{\boldsymbol{u}}(x_r)$. Given $\hat{\boldsymbol{u}}(x_r)$, we can design a steady-state generator with state $\theta(x_r) = [-x_{r1} \quad -x_{r2}]^{\mathrm{T}}$. Then based on (15), we derive matrices $\Phi = \begin{bmatrix} 0 & 1 \\ -1 & 1\end{bmatrix}$ and $\Psi = \begin{bmatrix} 1 & 0\end{bmatrix}$. After calculation, we have $\Phi_1 = \begin{bmatrix} 0 & 0 \\ 0 & -1\end{bmatrix}$ satisfying Assumption 4 and $\phi(x_r) = \Phi_1 a_1(x_r) = \begin{bmatrix} 0 & 0 \\ 0 & -1\end{bmatrix} x_{r1}^2$. The next step is to find the solution Ω of Sylvester equation (18). According to the property of Kronecker product, taking **vec** operation on both sides of (18) renders $\mathbf{vec}(\Omega\Phi) = (\Phi^{\mathrm{T}} \otimes I_2)\mathbf{vec}(\Omega)$ and $\mathbf{vec}(F\Omega) = (I_2 \otimes F)\mathbf{vec}(\Omega)$. Then Sylvester equation (18) is rewritten as

$$\begin{bmatrix}\Phi^{\mathrm{T}} \otimes I_2 & -I_2 \otimes F \\ -I_2 \otimes F & \Phi^{\mathrm{T}} \otimes I_2\end{bmatrix}\begin{bmatrix}\mathbf{vec}(\Omega) \\ \mathbf{vec}(\Omega)\end{bmatrix} = \begin{bmatrix}\mathbf{vec}(G\Psi) \\ \mathbf{vec}(G\Psi)\end{bmatrix}.$$

We let $F = \begin{bmatrix} -0.7 & 0 \\ 0 & -0.4\end{bmatrix}$ and $G = \begin{bmatrix} 0.549 \\ 1\end{bmatrix}$ such that F is nonsingular and the pair (F, G) is controllable. Then substituting system parameters into (18) and solving the above equation yields $\Omega = \begin{bmatrix} 0.42616 & -0.25068 \\ 0.89743 & -0.64102\end{bmatrix}$. After solving LMI conditions (27) and (28), we obtain the controller parameters for horizontal and vertical directions as follows:

$$A_p = \begin{bmatrix} 18.749 & -1292.58 & -486.05 & -623.56 \\ 38.51 & -2584.42 & -971.45 & -1246.23 \\ -78.76 & 5057.8 & 1899.11 & 2431.74 \\ -18.49 & 1218.39 & 459.07 & 589.91\end{bmatrix},$$
$$B_p = \begin{bmatrix} -2.86 & -0.000028 & 0.0002 & -0.000076\end{bmatrix}^{\mathrm{T}},$$
$$C_p = \begin{bmatrix} 115.27 & -7473.56 & -2810.15 & -3604.22\end{bmatrix}.$$

The controller of the double integrator in the horizontal direction is given by $u(t) = \Psi\Omega^{-1}\eta(t) + C_p x_p(t) + (1 - x_{r1}^2)x_{r2}$.

For the vertical direction, $C_r = [\ 0 \quad 1\ ]$, and the solution of regulator equation (14) is $\boldsymbol{u}(x_r) = -x_{r1} - 2x_{r1}x_{r2}^2 + x_{r1}^3 - 2x_{r1}^2 x_{r2} + x_{r1}^4 x_{r2}$. Using the same procedure, we obtain the same parameters $\Phi, \Psi, \Omega, A_p, B_p, C_p$, and the controller of the double integrator in the vertical direction is given by $u(t) = \Psi\Omega^{-1}\eta(t) + C_p x_p(t) - 2x_{r1}x_{r2}^2 + x_{r1}^3 - 2x_{r1}^2 x_{r2} + x_{r1}^4 x_{r2}$.

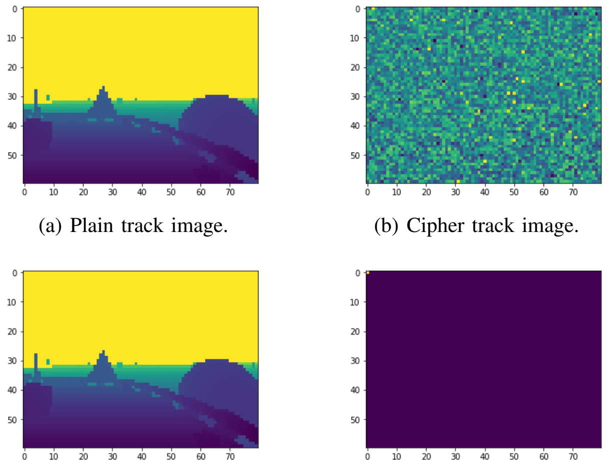
Lorenz's chaotic system is adopted for encryption [6]. Its T-S fuzzy model is written as (4) where $A_{e,1} = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & -30 \\ 0 & 30 & -\frac{8}{3}\end{bmatrix}$, and $A_{e,2} = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & 30 \\ 0 & -30 & -\frac{8}{3}\end{bmatrix}$. To transmit the image data, we pick $C_e = I_3$ and $B_e = \begin{bmatrix} 1 & 0 & 0\end{bmatrix}^{\mathrm{T}}$ such that $\mathrm{rank}(C_e B_e) = \mathrm{rank} B_e = 1$. After solving the LMI conditions (11), the gains of the state observer (8) are given by $L_{e,i} = \begin{bmatrix} 1.1497 & 0 \\ 0 & -0.5170\end{bmatrix}$, $i = 1, 2$.

C. Experiment results

1) Encryption: As [6], we evaluate the performance of chaotic encryption from the following two aspects: encryption and decryption speed, as well as key sensitivity.

Conventional encryption schemes usually involve modular exponentiation operations over large integers, which could be highly time-consuming, while our proposed chaotic encryption algorithm only involves simple matrix inverse operations and solves ordinary differential equations. For an 8-bit image in the experiments, the average speed of Paillier's algorithm [5] is about $1Kb/s$ with key length 1024, while the average speeds of our chaotic encryption and decryption are about $4.7Mb/s$ and $5.2Mb/s$, respectively. It demonstrates that the proposed chaotic encryption algorithm in this paper is significantly faster than the conventional encryption methods.

We let the initial state of the transmitter be the secret key Θ. Fig. 2 shows the chaotic encryption and decryption results of key sensitivity test. Specifically, Fig. 2(a) is a plain track image with size 80×60, Fig. 2(b) is its cipher track image, and Fig. 2(c) is the correctly recovered track image. The difference between the plain image and the correct decrypted image is 6.9282 where 2-norm is used. This demonstrates the correctness of chaotic encryption strategy. We assume that the attacker eavesdrops the cipher image z_e, and knows everything of the chaotic transmitter (4) except for the secret key, e.g., -10. If the eavesdropper instead uses -10.00000000001, the recovered image is a random image as shown in Fig. 2(d). The difference between the plain image and the incorrectly decrypted image is 3.4174×10^4. It shows that the cipher image cannot be accurately recovered with a slightly changed key, which demonstrates the key sensitivity of chaotic encryption.



(a) Plain track image.

(b) Cipher track image.

(c) Decrypted image (correct key).

(d) Decrypted image (wrong key).

Fig. 2. Results of chaotic encryption and decryption.

2) Path tracking in absence of evasion attacks: We choose an initial state as $(1.4\ 2.1)m$, which is at the center of the track. Fig. 3 shows the path tracking in first 15 seconds. Fig. 4(a) shows the tracking errors over time. In particular, dotted line depicts the tracking

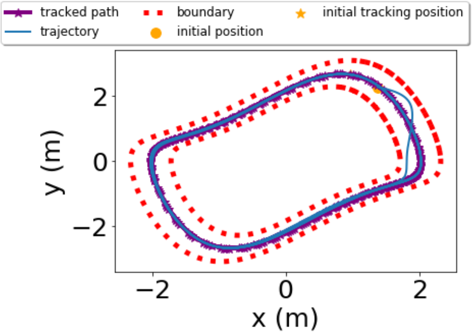
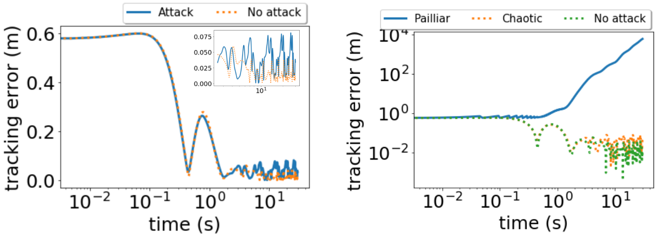


Fig. 3. Path tracking for the Van der Pol oscillator.



(a)

(b)

Fig. 4. Tracking error over time: (a) provides tracking error comparisons between attack-free scenario and attacked scenario; (b) provides tracking error comparisons applying chaotic encryption and Paillier algorithm.

error in absence of evasion attacks, and the subfigure clearly shows that the steady tracking error is smaller than $0.05m$ where 2-norm is used, and the settling time is about 2 seconds. Fig. 4(a) demonstrates that the double integrator can quickly track the path with a small steady-state error despite the learning error of the perception mapping p.

We replace chaotic encryption with partially homomorphic encryption algorithm, e.g., Paillier algorithm [5], in the control loop. In terms of high security, the key length of Paillier algorithm is typically chosen as 1024 [25]. The controller performance comparisons applying chaotic encryption and Paillier algorithm are shown in Fig. 4(b). It can be seen that the tracking error diverges using 1024-bits paillier algorithm since time delay caused by encryption and decryption is introduced in the control loop.

3) Path tracking under evasion attacks: We adopt the fast gradient sign method (FGSM) to generate the attacks. As [19], the perturbation of a plain image is denoted as $d \triangleq c_0 \text{sign}(\nabla_{z_i} J)$ where J is the cost function of the trained model, ∇_{z_i} is the gradient of the model and c_0 is the perturbation's amplitude. Here the attacker chooses $c_0 = 5$. Then the ith corrupted image is given by $\check{z}_i = z_i + d_i$.

We choose the same initial state as $(1.4\ 2.1)m$. Solid line in Fig. 4(a) shows the tracking errors subject to evasion attacks over time. The steady tracking error is smaller than $0.08m$, and the settling time is also about 2 seconds. The simulation results demonstrate that evasion attacks only induce small degradation of tracking performance.

V. Conclusion

In this paper, we study perception-driven control of a mobile robot for path tracking. We consider passive attacks on image transmission and evasion attacks on a machine learning system. To defeat the passive attacks, we utilize chaotic encryption technique to mask pixels of plain images in real time, and construct a bank of fuzzy unknown input observers to decrypt the cipher pixels in parallel. As for the path tracking, we design a robust output-feedback controller, which can attenuate the effects of the evasion attacks and learning errors of the machine learning system by reducing $\mathcal{L}_2$ gain. Simulations are conducted in the CARLA simulator to demonstrate robust path tracking and secure image transmission.

References

[1] T. Litman, "Autonomous vehicle implementation predictions: Implications for transport planning," Victoria, British Columbia, Canada, 2013.

[2] Y. Xun, J. Liu, N. Kato, Y. Fang, and Y. Zhang, "Automobile driver fingerprinting: A new machine learning based authentication scheme," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1417-1426, 2020.

[3] J. Kim and H. Shim, "Encrypted state estimation in networked control systems," *Conference on Decision and Control*, Nice, France, pp. 7190-7195, 2019.

[4] J. H. Cheon, K. Han, H. Kim, J. Kim, and H. Shim, "Need for controllers having integer coefficients in homomorphically encrypted dynamic system," *Conference Decision Control*, Miami Beach, USA, pp. 5020–5025, 2018.

[5] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *International Conference on the Theory and Applications of Cryptographic Techniques*, Prague, Czech Republic, pp. 223-238, 1999.

[6] K.-Y. Lian, C.-S. Chiu, T.-S. Chiang, and P. Liu, "Secure communications of chaotic systems with robust performance via fuzzy observer-based design," *IEEE Transactions on Fussy Systems*, vol. 9, no. 1, pp. 212-220, 2001.

[7] M. B. Jensen, K. Nasrollahi, and T. B. Moeslund, "Evaluating state-of-the-art object detector on challenging traffic light data," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Honolulu, HI, pp. 882-888, 2017.

[8] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," *2013th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III*, pp. 387-402, 2013.

[9] L. Zhou, V. Tzoumas, G. J. Pappas, P. Tokekar. "Distributed attack-robust submodular maximization for multi-robot planning," *IEEE International Conference on Robotics and Automation*, Paris, France, pp. 2479-2485, 2020.

[10] P. Guo, H. Kim, N. Virani, J. Xu, M. Zhu and P. Liu, " RoboADS: Anomaly detection against sensor and actuator misbehaviors in mobile robots," *IEEE/IFIP International Conference on Dependable Systems and Networks*, Luxembourg City, pp. 574-585, 2018.

[11] M. Hou and P. C. Müller, "Design of observer for linear systems with unknown inputs," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 871-875, 1992.

[12] X. Zhang, Z. Yuan, S. Xu, Y. Lu, and M. Zhu, "Secure perception-driven control of mobile robots using chaotic encryption," *American Control Conference*, New Orleans, LA, USA, pp. 2575-2580, 2021.

[13] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust guarantees for perception-based control," *Conference on Learning for Dynamics and Control*, on-line, pp. 350-360, 2020.

[14] W. Böhmer, S. Grünewälder, Y. Shen, M. Musial, and K. Obermayer, "Construction of approximation spaces for reinforcement learning," *Journal of Machine Learning Research*, vol. 14, pp. 2067-2118, 2013.

[15] H. Sekkat, S. Tigani, R. Saadane, and A. Chehri, "Vision-based robotic arm control algorithm using deep reinforcement learning for autonomous objects grasping," *Applied Sciences*, vol. 11, no. 7917, pp. 1-14, 2021.

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," *Conference on Robot Learning*, Mountain View, CA, USA, pp. 1-16, 2017.

[17] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation ax + xb = c," *Communications of the ACM*, vol. 15, no. 9, pp. 820-826, 1972.

[18] Z. Chen and J. Huang, "Robust output regulation with nonlinear exosystems," *Automatica*, vol. 41, pp. 1447-1454, 2005.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *International Conference on Learning Representations*, San Diego, CA, USA, pp. 1-11, 2015.

[20] G. Millérioux, J. M. Amigó, and J. Daafouz, "A connection between chaotic and conventional cryptography," *IEEE Transactions on Circuit and Systems I: Regular Papers*, vol. 55, no. 6, pp. 1695-1703, 2008.

[21] R. M. Murray, M. Rathinam, and W. Sluis, “Differential flatness of mechanical control systems: A catalog of prototype systems,” *Proceedings of the ASME International Mechanical Engineering Congress and Exposition*, San Francisco, CA, USA, pp. 1-9, 1995.

[22] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. “Multilayer feedforward networks with non-polynomial activation functions can approximate any function,” *Neural Networks*, vol. 6, no. 6, pp. 861-867, 1993.

[23] J. Li, Y. Yang, J. S. Sun, K. Tomsovic and H. Qi, “Conaml: Constrained adversarial machine learning for cyber-physical systems,” *Proceedings of ACM Asia Conference on Computer and Communications Security*, pp. 52-66, 2021.

[24] Z. Zhang, M. Sun, R. Deng, C. Kang, and M.-Y. Chow, “Physics-constrained robustness verification of intelligent security assessment for power systems,” *IEEE Transactions on Power Systems*, pp. 1-13, 2022.

[25] J. Christine, L. Ha, M. Alexander, and S. Ben, “Encryption performance improvements of the Paillier cryptosystem,” *https://eprint.iacr.org/2015/864.pdf*, 2015.

[26] G. Balas, R. Chiang, A. Packard, and M. Safonov, “Robust Control Toolbox User’s Guide,” https://ostad.nit.ac.ir/payaidea/ospic/file1502.pdf, 2015.

[27] X. Zhang, Z. Yuan, S. Xu, Y. Lu, and M. Zhu, “Secure perception-driven control of mobile robots using chaotic encryption,” https://www.dropbox.com/s/g17cdkii2s9jq8b/complete%20version.pdf?dl=0, 2023.