2021 American Control Conference (ACC)
New Orleans, USA, May 25-28, 2021

Secure perception-driven control of mobile robots using chaotic encryption

Xu Zhang, Zhenyuan Yuan, Siyuan Xu, Yang Lu and Minghui Zhu

Abstract— **This paper considers perception-driven control of a mobile robot for reference tracking where perception is performed by a machine learning system. The robot is subject to passive attacks and evasion attacks on image transmission. A robust output feedback controller together with a chaotic encryption system ensures input-to-state stability of the closed-loop system, and the chaotic encryption approach keeps image transmission secure. Simulations are conducted in the CARLA simulator to demonstrate robust reference tracking and secure image transmission.**

I. INTRODUCTION

Mobile robotic systems are becoming ubiquitous and rapidly evolving in many areas, e.g., unmanned aerial vehicles and self-driving cars [1]. Mobile robots integrate heterogeneous devices for embedded sensing, mobile computing and real-time control. These devices exchange information via on-board communication medium. For example, modern urban vehicles carry as many as 150 electronic control units (ECUs), which communicate with each other by intra-vehicle component communication protocol, e.g., Control Area Network (CAN) bus [2]. Moreover, machine learning techniques have been increasingly applied to improve intelligence of mobile robots. For example, deep learning is implemented in almost every aspect of autonomous driving, e.g., driving scene perception, localization and path planning [3].

However, a wide spectrum of privacy and security issues arise and mobile robots are severely threatened by various intentional attacks. Such adversaries cause privacy breach, induce undesired operations and give rise to misbehavior. This paper specifically considers two classes of attacks. One is passive attack on intra-robot communication, which can be launched to eavesdrop important information during data transmission. For example, an attacker is able to fingerprint drivers with extremely high accuracies by collecting intra-vehicle sensor data [4]. The other one is active attack against machine learning algorithms at test time. For example, an attacker is able to adversarially manipulate a small number of pixels for a stop sign image, thus as unexpected, the modified image could be misclassified as a speedlimit sign by the deep neural network [5].

Literature review. Cryptography is a standard practice to ensure data security during transmission. This paper considers secure transmission of real-time image data for perception. Due to inherent features of images, e.g., bulky data size and high correlation among pixels, conventional encryption algorithms such as DES and RSA are not suitable for practical image encryption, especially when encryption needs to be done in real time. The reason is that it is rather difficult for conventional encryption algorithms to swiftly shuffle and diffuse image data. Since chaotic systems are extremely sensitive to initial states and system parameters, chaotic-based encryption methods can provide exceptionally good properties with regard to strong security and high speed, especially on encrypting high-dimensional image data [6]. Currently, there are two strategies for chaotic encryption implementation. One strategy is to combine plain images with different chaotic maps or their combinations. An incomplete list of references includes [7], [8]. These papers experimentally demonstrate strong security as well as fast encryption speed. The second strategy is to inject plain images into a chaotic system. Paper [10] used the Takagi-Sugeno fuzzy model to represent chaotic systems, and developed a fuzzy observer to recover plain images. Paper [11] developed a multi-observer approach to decrypt cipher images encrypted by a nonlinear chaotic system. In this set of papers, chaotic encryption is not integrated with control problems of dynamic systems.

Adversarial machine learning has been receiving increasing attention in the area of artificial intelligence. Typical attacks on machine learning systems include poisoning attacks and evasion attacks. Poisoning attacks manipulate training data to cause misclassification while evasion attacks tamper test data to evade a trained classifier at test time. Please refer to survey paper [5] for detailed discussion on adversarial machine learning. This set of results do not consider mitigation of the attacks on control systems.

There have been recent works which study attack-resilient estimation and control of robot systems. Confidentiality, integrity, and availability, known as the CIA triad, is the classic categorization of information security. Denial-of-Service (DoS) and replay attacks compromise data availability and are entailed in papers [12] [13]. Sensor attacks compromise data integrity and are studied in papers [14], [15], [16]. In [14], the authors proposed a stochastic strategy for motion planning of unmanned aerial vehicles subject to sensor attacks. Based on the redundancy of sensor measurements, the authors in [15] proposed a mitigation strategy where the attack-resilient estimator can identify attacks and the controller can drive ground robots under sensor attacks to reach a desired state without being hijacked. Paper [16] designed a secure observer-based distributed controller such that a group

The authors are with the School of Electrical Engineering and Computer Science, Pennsylvania State University xxz313@psu.edu, zqy5086@psu.edu, spx5032@psu.edu, yml5046@psu.edu, muz16@psu.edu

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

2021 American Control Conference (ACC) | 978-1-6654-4197-1/21/$31.00 ©2021 IEEE | DOI: 10.23919/ACC50511.2021.9483382

of vehicles subject to sensor attacks can achieve formation control. However, the aforementioned papers do not take into account confidentiality of transmitted data, especially high-dimensional data. Moreover, the above papers on sensor attacks impose the assumption on the maximum number of attacked sensors. This assumption is not required in the current paper.

Contributions. In this paper, we consider perception-driven control of a mobile robot for reference tracking where perception is performed by a machine learning system. The robot is subject to passive attacks on image transmission and evasion attacks on the machine learning system. To defeat passive attacks, the camera injects real-time plain images into a chaotic system represented by Takagi-Sugeno fuzzy system. The perception unit uses an unknown input observer to decrypt the cipher images. A robust output-feedback controller leverages decrypted images to ensure input-to-state stability of the closed-loop system despite the attacks and learning errors of the machine learning system. The CARLA platform is used to conduct simulations on double integrator. The simulation results demonstrate robust reference tracking and secure image transmission.

Paper organization. Section II introduces the secure perception-driven control problem of a linear time-invariant system. In Section III, we disguise the image data by using a chaotic encryption method and develop an unknown input observer to decrypt the cipher images. Also, we design a robust output-feedback controller to realize reference tracking despite learning errors and evasion attacks. Proofs are provided in Section IV. In Section V, we use a double integrator to demonstrate the algorithm performance.

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 real $m\times n$ matrices. The set of m-dimensional symmetric positive definite matrices is denoted by $\mathbb{S}_+^m$. A block diagonal matrix with submatrices $X_1,\ldots,X_p$ on 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 and the hermitian operator $He\{\cdot\}$ is defined as $He\{\Gamma\}\triangleq\Gamma+\Gamma^{\mathrm{T}}$. An 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 $\lambda_{\min}(\Gamma)$ 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. For function $u(t):[0,\infty)\to\mathbb{R}^n$, we denote the supremum norm of the truncation of $u(t)$ in $[t_1,t_2]$ by $\|u_{[t_1,t_2]}\|\triangleq\sup_{t_1\le t\le t_2}\|u(t)\|$. We also denote $\|u\|_a\triangleq\limsup_{t\to\infty}\|u(t)\|$. For function $x(t):[0,\infty)\to\mathbb{R}^n$, its 2-norm is denoted by $\|x(t)\|_2=\left(\int_0^\infty x(t)^{\mathrm{T}}x(t)dt\right)^{1/2}$. A function $\gamma:\mathbb{R}_+\to\mathbb{R}_+$ is of class $\mathcal{K}$ if it is continuous, positive definite, 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 $t\ge0$, $\beta(s,t)$ is of class $\mathcal{K}$ and for each fixed $s\ge0$, $\beta(s,t)$ decreases to zero as $t\to\infty$. Function composition is defined by $g\circ f(x)\triangleq g(f(x))$. Consider a nonlinear system

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

where $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^n$ is locally Lipschitz continuous in x and u with $f(0,0)=0$, $h:\mathbb{R}^n\to\mathbb{R}^l$ is continuous with $h(0)=0$ and $u(t)$ is a piecewise continuous, bounded function for all $t\ge0$.

Definition 1: [23] System (1) is said to be input-to-state stable (ISS) from $u(t)$ to $x(t)$ if there exist a class $\mathcal{KL}$ function β and a class $\mathcal{K}_\infty$ function γ such that for any initial state $x(t_0)$ and any input $u(t)$, $\|x(t)\|\le\beta(\|x(t_0)\|,t-t_0)+\gamma\left(\|u_{[t_0,t]}\|\right)$.

Definition 2: [23] System (1) satisfies asymptotic gain (AG) from $u(t)$ to $x(t)$ if there exists a class $\mathcal{K}_\infty$ function γ such that for any initial state $x(t_0)$ and input $u(t)$, $\|x\|_a\le\gamma\left(\|u\|_a\right)$.

Definition 3: [24] System (1) is input-to-output stable (IOS) from $u(t)$ to $y(t)$ 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 input $u(t)$, $\|y(t)\|\le\beta(\|x(t_0)\|,t-t_0)+\gamma\left(\|u_{[t_0,t]}\|\right)$.

Definition 4: [24] System (1) satisfies the output-asymptotic $\mathcal{L}_\infty$ stability (oALS) property from $u(t)$ to $y(t)$ if there exists a class $\mathcal{K}$ function γ so that for all $x(t_0)$ and input $u(t)$, $\|y\|_a\le\gamma\left(\|u\|_a\right)$.

II. Problem formulation

This section introduces secure perception-driven control of a mobile robot using chaotic encryption. This paper only considers secure transmission of high-dimensional images. There are many efficient encryption schemes, e.g., DES, AES, IDEA [7], for secure transmission of low-dimensional data, e.g., positions and heading of a rear-wheel-drive vehicle. So it is not considered in this paper.

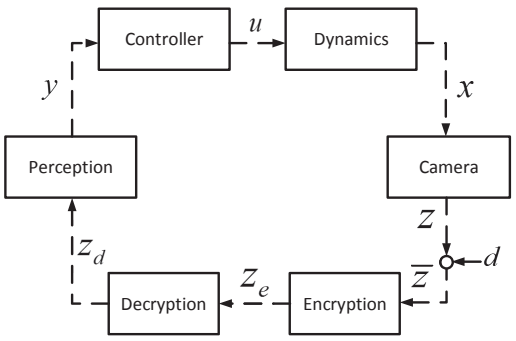


Fig. 1. Feedback loop of perception-driven control.

A. System model without encryption and evasion attack

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

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

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 [25], 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 extracted information $y(t)$ is used to generate control command $u(t)$. Given a desired output matrix C, the perception mapping p is learned from training data set $\{(z(t), Cx(t))\}$ such that $p \circ q(x(t)) = Cx(t)$. Due to inherent learning errors, $p \circ q(x(t))$ may not be represented by $Cx(t)$ exactly. Then the output equation becomes $y(t) = Cx(t) + w(t)$ where $w(t) \triangleq p \circ q(x(t)) - Cx(t)$ describes the learning error of mapping p.

B. System model with encryption and evasion attack

The image data can be tampered by evasion attack $d(t) \in \mathbb{R}^{n_p}$. The image is free of evasion attack at time instant t if $d(t) = 0_{n_p}$, otherwise the evasion attack can take any value from -255 to 255 to alter the pixels at test time. We denote the attacked image vector by $\bar{z}(t) \triangleq z(t)+d(t)$, 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 attacked plain image and the corresponding attacked cipher image is represented as $z_e(t) \triangleq \mathscr{E}_\Theta(\bar{z}(t))$. 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 $\bar{z}_d(t) \triangleq \mathscr{D}_\Theta(z_e(t))$.

The decrypted image $\bar{z}_d(t)$ passes through the perception unit and thus $y(t) = p(\bar{z}_d(t))$. By function composition, we get $y(t) = p \circ \mathscr{D}_\Theta \circ \mathscr{E}_\Theta \circ (q(x(t)) + d(t))$. This case implies

$$\dot{x}(t) = Ax(t) + Bu(t),\ y(t) = Cx(t) + w'(t) \qquad (3)$$

where $w'(t) \triangleq p \circ \mathscr{D}_\Theta \circ \mathscr{E}_\Theta \circ (q(x(t)) + d(t)) - p \circ q(x(t)) + w(t)$. Note that the first difference term represents the error caused by decryption and evasion attacks.

C. Control objective and assumptions

We aim to design a controller for system (3) such that output $y(t)$ can keep track of reference input $r(t) \in \mathbb{R}^l$.

Assumption 1: The reference signal $r(t)$ and learning error $w(t)$ are differentiable; both $w(t)$ and $\dot{w}(t)$ are uniformly upper bounded.

Assumption 2: Mappings p and q are continuously differentiable.

Assumption 3: The dimension of input and output satisfies $l = m$ and $\text{rank}(CB) = m$.

Assumption 4: (C, A) is observable.

An example to satisfy Assumption 3 is single-input-single-output systems where $l = m = 1$ and $\text{rank}(CB) = 1$. Assumption 4 ensures the state observer of system (3) exists.

Our objective is to ensure real-time and secure high-dimensional image data transmission and attenuate the effects of learning error $w(t)$ and evasion attack $d(t)$ on reference tracking. It will be achieved via a robust output-feedback controller and chaotic encryption.

Remark 1: For ease of presentation, this paper only uses camera for perception. Our results are directly applicable to other sensors; e.g., LiDAR and RADAR, whose data is also high-dimensional.

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 tracking controller.

A. Chaotic encryption

This section employs message-embedding chaotic encryption methods, and injects the plain image into a chaotic system. For implementation convenience in control theory, many chaotic systems, e.g., Lorenz's system and Chua's circuit [18], can be written as the following Takagi-Sugeno (T-S) fuzzy system

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

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, the scheduling variable $\xi(t)$ includes part of system state $x_e(t)$, and the output $z_e(t)$ includes the scheduling variable $\xi(t)$ as a subvector; $\sum_{i=1}^N \mu_i(\xi(t)) = 1$ and $0 \le \mu_i(\xi(t)) \le 1$.

The camera maintains n_p chaotic transmitters that are in the same form and executed in parallel. In particular, each pixel $\bar{z}^{(j)}(t)$ of attacked plain image $\bar{z}(t)$ is injected into the chaotic transmitter (4), which becomes

$$\begin{aligned} \dot{x}_e^j(t) &= \textstyle\sum_{i=1}^N \mu_i(\xi^j(t)) \left(A_{e,i}x_e^j(t) + B_{e,i}\bar{z}^{(j)}(t)\right) \\ z_e^{(j)}(t) &= \textstyle\sum_{i=1}^N \mu_i(\xi^j(t))C_{e,i}x_e^j(t). \end{aligned} \qquad (5)$$

In above, the pixel $\bar{z}^{(j)}(t)$ acts as an unknown input of the jth transmitter where state is $x_e^j(t) \in \mathbb{R}^{n_s}$ and output is $z_e^{(j)}(t) \in \mathbb{R}^{n_e^j}$. For simplicity, all the transmitters share the same input matrix B_e and output matrix C_e. In cryptography, $x_e^j(t)$ is also called the keystream. For the jth transmitter, the initial state $x_e^j(0)$ 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 $z_e^{(j)}(t)$ and $\dot{z}_e^{(j)}(t)$ to recover the attacked plain pixel $\bar{z}^{(j)}(t)$ (unknown input). This will be achieved via an unknown input observer (UIO).

Assumption 5: The perception unit can access $\dot{z}_e^{(j)}(t)$.

Assumption 6: $\text{rank}(C_eB_e) = \text{rank}B_e = 1$.

Assumption 5 is used to guarantee the recovery of the unknown input, and it is a standard and necessary assumption for the continuous-time UIO technique (see [19]). Assumption 6 is needed to conduct the following transformation on the state and output. As [17], we decompose system (5) into two subsystems: one is free of the unknown input, and the other is dependent on it. Matrices $T_e \in \mathbb{R}^{n_s \times n_s}$ and $U_e \in \mathbb{R}^{n_e^j \times n_e^j}$ are defined as follows:

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

Note that T_e is nonsingular since $T_e T_e^{-1} = I_{n_s}$ 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 decomposition $\bar{x}_e^j \triangleq T_e x_e^j$ and output decomposition $\bar{z}_e^{(j)} \triangleq U_e z_e^{(j)}$, system (5) is partitioned into a new form

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

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

$$\begin{aligned}
A_{e,i}^1 &= B_e^\perp A_{e,i} \left[I_{n_s} - B_e(C_eB_e)^\dagger C_e\right](B_e^\perp)^\dagger \\
A_{e,i}^2 &= B_e^\perp A_{e,i} B_e \\
A_{e,i}^3 &= (C_eB_e)^\dagger C_e A_{e,i} \left[I_{n_s} - B_e(C_eB_e)^\dagger C_e\right](B_e^\perp)^\dagger \\
A_{e,i}^4 &= (C_eB_e)^\dagger C_e A_{e,i} B_e \\
\bar{C}_e &= (C_eB_e)^\perp C_e \left[I_{n_s} - B_e(C_eB_e)^\dagger C_e\right](B_e^\perp)^\dagger.
\end{aligned}$$

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

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

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

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

The perception unit uses the UIO below to recover $\bar{z}^{(j)}(t)$

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

where $\hat{\bar{x}}_{e,1}^j(t)$ is the estimate of $\bar{x}_{e,1}^j(t)$, $L_{e,i}$, $i = 1, \ldots, N$ are the observer gains, and $\bar{z}_d^{(j)}(t)$ is the decrypted pixel. Recall that $\xi^j(t)$ and $\dot{z}_e^{(j)}(t)$ are available to the perception unit and thus the UIO (8). We define the estimation error as $\tilde{\bar{x}}_{e,1}^j(t) \triangleq \bar{x}_{e,1}^j(t) - \hat{\bar{x}}_{e,1}^j(t)$, then we derive the error dynamics

$$\dot{\tilde{\bar{x}}}_{e,1}^j(t) = \textstyle\sum_{i=1}^N \mu_i(\xi^j(t)) \left((A_{e,i}^1 - L_{e,i}\bar{C}_e)\tilde{\bar{x}}_{e,1}^j(t)\right). \quad (9)$$

The following lemma employs a common Lyapunov function to derive a sufficient condition for exponential convergence of estimation errors.

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

$$\begin{aligned}
&(A_{e,i}^1)^\mathrm{T} P_e + P_e A_{e,i}^1 - (\bar{C}_e)^\mathrm{T} Q_{e,i}^\mathrm{T} - Q_{e,i}\bar{C}_e + \gamma_e I < 0, \\
&\forall i = 1, \ldots, N,
\end{aligned} \quad (10)$$

the error dynamics (9) is globally exponentially stable with observation gain matrices $L_{e,i} = P_e^{-1} Q_{e,i}$, and $\|\bar{z}(t) - \bar{z}_d(t)\|$ diminishes exponentially.

B. Robust tracking controller

To realize trajectory tracking, we first derive tracking error dynamics. By Assumption 1, we differentiate tracking error $e(t) \triangleq y(t) - r(t)$ with regard to time t, and substituting it into (3) yields $\dot{e}(t) = CAx(t) + CBu(t) + \dot{w}'(t) - \dot{r}(t)$. By Assumption 3, the controller is represented as

$$u(t) = (CB)^{-1}\left(-K_1 e(t) - CA\hat{x}(t) + \dot{r}(t)\right) + u_e(t) \quad (11)$$

where $\hat{x}(t)$ is the estimate of the state $x(t)$, $u_e(t)$ is a robust controller and K_1 is a positive definite matrix. We denote the state estimation error by $\tilde{x}(t) \triangleq \hat{x}(t) - x(t)$. Then the augmented error dynamics is compactly written as follows:

$$\begin{aligned}
\dot{\bar{x}}(t) &= \bar{A}\bar{x}(t) + \bar{B}_0 u_e(t) + \bar{B}_1 \Delta(t) + \bar{B}_1 \bar{w}(t) \\
\bar{y}(t) &= \bar{C}\bar{x}(t)
\end{aligned} \quad (12)$$

where $\bar{x}(t) \triangleq [e(t)^\mathrm{T}\ \tilde{x}(t)^\mathrm{T}]^\mathrm{T}$, $\bar{w}(t) \triangleq \begin{bmatrix} w(t) \\ \dot{w}(t) \end{bmatrix}$, $\Delta(t) \triangleq \begin{bmatrix} p(\bar{z}_d(t)) - p(z(t)) \\ \dot{p}(\bar{z}_d(t)) - \dot{p}(z(t)) \end{bmatrix}$, $\bar{A} \triangleq \begin{bmatrix} -K_1 & -CA \\ 0 & A - LC \end{bmatrix}$, $\bar{B}_0 \triangleq \begin{bmatrix} CB \\ 0 \end{bmatrix}$, $\bar{B}_1 \triangleq \begin{bmatrix} 0 & I_l \\ L & 0 \end{bmatrix}$, $\bar{C} \triangleq \begin{bmatrix} I_l & 0 \end{bmatrix}$, and L is a full-order observer gain vector. Note that the evasion attack $d(t)$ is included in $\Delta(t)$. In particular, if $w(t) = 0$ and $\bar{z}_d(t) = z(t)$, then there is no need to add the extra control $u_e(t)$ into (11), i.e., $u_e(t) = 0$. By Assumption 4, there exists a matrix L such that $A - LC$ is Hurwitz, which implies that $\lim_{t\to\infty} e(t) = 0$ and $\lim_{t\to\infty} \tilde{x}(t) = 0$.

For system (12), we will design a robust output-feedback reference tracking controller such that the effects of $\bar{w}(t)$ on $e(t)$ are attenuated. The reference signal $r(t)$ is known and so are $e(t)$ and $\bar{y}(t)$. Then, the tracking error $e(t)$ can be directly fed into the following dynamic compensator

$$\begin{aligned}
\dot{x}_c(t) &= A_c x_c(t) + B_c \bar{y}(t) \\
u_e(t) &= C_c x_c(t) + D_c \bar{y}(t)
\end{aligned} \quad (13)$$

with $x_c(t) \in \mathbb{R}^{n_c}$. Matrices A_c, B_c, C_c, D_c are controller parameters. After substituting $u_e(t)$ in (13) into (12) and augmenting $\bar{x}(t)$ by $x_c(t)$, the system is written as

$$\begin{aligned}
\dot{\bar{x}}'(t) &= \bar{A}'\bar{x}'(t) + \bar{B}'\Delta(t) + \bar{B}'\bar{w}(t) \\
e(t) &= \bar{C}'\bar{x}'(t) + \bar{D}'\bar{w}(t)
\end{aligned} \quad (14)$$

where $\bar{x}'(t) \triangleq \left[\bar{x}(t)^\mathrm{T}\ x_c(t)^\mathrm{T}\right]^\mathrm{T}$, and

$$\begin{aligned}
&\bar{A}' \triangleq \begin{bmatrix} \bar{A} + \bar{B}_0 D_c \bar{C} & \bar{B}_0 C_c \\ B_c \bar{C} & A_c \end{bmatrix}, \bar{B}' \triangleq \begin{bmatrix} \bar{B}_1 \\ 0 \end{bmatrix}, \\
&\bar{C}' \triangleq \begin{bmatrix} \bar{C} & 0 \end{bmatrix}, \bar{D}' \triangleq 0_{l\times 2l}.
\end{aligned}$$

The following theorem gives a sufficient condition on

stability of system (14), and derives the output-feedback controller parameters A_c, B_c, C_c, D_c.

Theorem 1: Under Assumptions 1, 2, 3 and 4, if there exist $R \in \mathbb{S}_+^{n+l}$, $S \in \mathbb{S}_+^{n+l}$, and rectangular matrices $\hat{A}_c$, $\hat{B}_c$, $\hat{C}_c$, $\hat{D}_c$, and scalar $\gamma_c \in \mathbb{R}_+$ such that the following LMIs are feasible

$$\begin{bmatrix} R & I_{n+l} \\ I_{n+l} & S \end{bmatrix} > 0, \tag{15}$$

$$\begin{bmatrix} He\{\bar{A}R + \bar{B}_0\hat{C}_c\} + R & \star & \star & \star \\ \hat{A}_c + \bar{A}^{\mathrm{T}} + \bar{C}^{\mathrm{T}}\hat{D}_c^{\mathrm{T}}\bar{B}_0^{\mathrm{T}} + I_{n_c} & He\{S\bar{A} + \hat{B}_c\bar{C}\} + S & \star & \star \\ \bar{B}_1^{\mathrm{T}} & \bar{B}_1^{\mathrm{T}}S & -(\gamma_c - 1)I_{2l} & \star \\ \bar{C}R & \bar{C} & 0 & -(\gamma_c - 1)I_l \end{bmatrix} < 0, \tag{16}$$

then system (14) satisfies the following four properties:

1) ISS from $\Delta(t)$ and $\bar{w}(t)$ to $\bar{x}'(t)$;
2) AG from $\bar{w}(t)$ to $\bar{x}'(t)$;
3) oALS from $\bar{w}(t)$ to $e(t)$;
4) $\mathcal{L}_2$ gain from $\Delta(t)$ and $\bar{w}(t)$ to $e(t)$ less than γ_c.

Additionally, the parameters A_c, B_c, C_c, D_c, in controller (13) can be computed by

$$\begin{bmatrix} A_c & B_c \\ C_c & D_c \end{bmatrix} = \begin{bmatrix} Y & S\bar{B}_0 \\ 0 & I_l \end{bmatrix}^{-1} \times \left(\begin{bmatrix} \hat{A}_c & \hat{B}_c \\ \hat{C}_c & \hat{D}_c \end{bmatrix} - \begin{bmatrix} S\bar{A}R & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} M^{\mathrm{T}} & 0 \\ \bar{C}R & I_l \end{bmatrix}^{-1} \tag{17}$$

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

IV. SIMULATION

This section provides a simulation by using a double integrator in the CARLA simulator [26]. The computer used in the simulation is Core $i7-3632$ QM CPU with 2.20 GHz and 15.5 GiB Memory. The dynamics of the double integrator for x-coordinate is given by

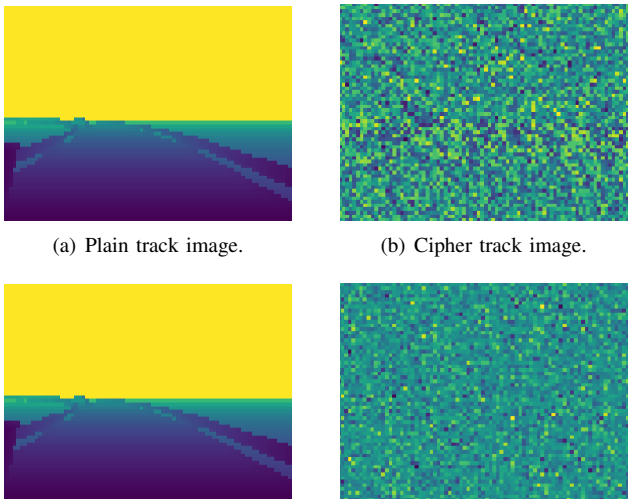
$$\dot{\phi}_x(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \phi_x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \psi_u(t),$$

where ϕ_x includes position and velocity and ψ_u is acceleration. Note that y-coordinate follows the same model as x-coordinate, and its state is denoted by $\phi_y(t)$. The output equation is $z(t) = q(\phi_x(t), \phi_y(t))$ where $z(t)$ is the plain image vector. The double integrator moves insides a track with width 10m and aims to follow the center of the track, a circle centered at $(0, 0)$ with radius 150m.

A. Encryption and decryption

Lorenz's chaotic system is adopted for encryption [18]. Its T-S fuzzy model is written as (4) where $A_{e,1} = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & -30 \\ 0 & 30 & -\frac{8}{3} \end{bmatrix}, A_{e,2} = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & 30 \\ 0 & -30 & -\frac{8}{3} \end{bmatrix}$. We inject each pixel of attacked plain image into the chaotic transmitter (4). In particular, we choose the mixing matrix $B_e = [1, 1, 1]^{\mathrm{T}}$ and let $C_e = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ match the individual equation (5). It is clear that $\mathrm{rank}(C_eB_e) = \mathrm{rank}B_e = 1$ and Assumption 6 is satisfied.

The average encryption speed and decryption speed are 2.8 MB/s and 10 MB/s, respectively. It shows that the chaotic encryption and decryption speeds are fast. Notice that 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 dynamics. In addition, Fig. 2 shows the performance of chaotic encryption and decryption. Specifically, Fig. 2(a) is a plain track image with size 288×214, Fig. 2(b) is its cipher track image, and Fig. 2(c) is the correctly recovered track image. This demonstrates the correctness of chaotic encryption strategy. We assume that the attacker eavesdrops the cipher image $z_{e,1}$, and knows everything of the chaotic transmitter (4) except for value -10 in $A_{e,1}$. If the eavesdropper instead uses -10.000001, the recovered image is a random image as shown in Fig. 2(d). This demonstrates the security of chaotic encryption strategy.



(a) Plain track image. (b) Cipher track image.

(c) Decrypted image (correct key). (d) Decrypted image (wrong key).

Fig. 2. Performance of chaotic encryption and decryption.

B. Path tracking

Let the output matrix be $C = [1\ 1]$ such that Assumptions 3-4 are satisfied. Given the training data $\{(z(t), Cx(t))\}$, we learn the perception mapping p by convolutional neural network (CNN) [27]. Assume $K_1 = 5$ and the observer gain is $L = \begin{bmatrix} -84 \\ 90 \end{bmatrix}$. We solve the LMIs (15) and (16) and obtain the controller parameters

$$C_c = \begin{bmatrix} -3.19 & 456.79 & -473.37 \end{bmatrix}, D_c = \begin{bmatrix} -129.4 \end{bmatrix},$$
$$A_c = \begin{bmatrix} 93.6 & 1826.8 & -2016.99 \\ 1.17 & -254.7 & 262.4 \\ 2.3 & -221.79 & 226.6 \end{bmatrix}, B_c = \begin{bmatrix} 3168.3 \\ 54.45 \\ 85.05 \end{bmatrix}.$$

Moreover, $\mathcal{L}_2$ gain is computed by $\gamma_c = 1.000053$.

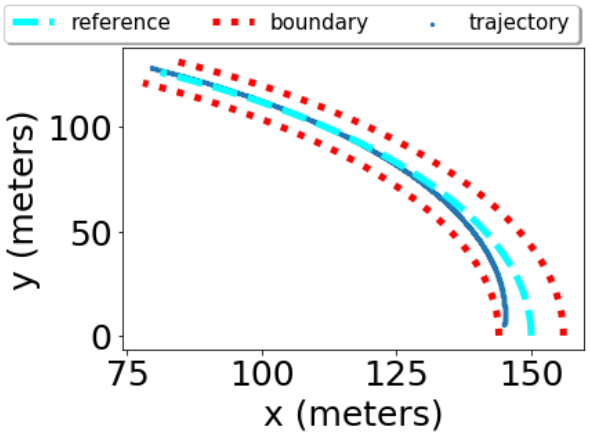


Fig. 3. Initial portion of reference tracking.

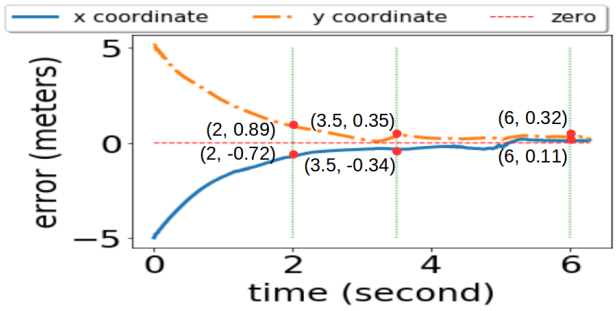


Fig. 4. Tracking error.

We choose the initial state as $(145, 5)$, which is on the boundary of the track. Fig. 3 shows the reference tracking in about one second. Fig. 4 shows the tracking errors along x, y-coordinate over time. The steady tracking error is restricted within 0.35m, and the settling time is about 2s. Figs. 3 and 4 demonstrate that the double integrator can quickly track the circle with a small steady-state error despite the learning error of the perception mapping p.

V. Conclusion

We study secure perception-driven control of a mobile robot for reference tracking where perception is performed by a machine learning system. Chaotic encryption strategy is employed to disguise the image data and ensure secure transmission, and a robust output-feedback controller is developed to realize reference tracking. The performance of the proposed algorithm is demonstrated by a double integrator in the CARLAR simulator.

References

[1] T. Litman. Autonomous vehicle implementation predictions: Implications for transport planning. `https://www.vtpi.org/avip.pdf`. 2020.

[2] M. E. Harb and A. M. Elshaer. Autonomous car implementation based on CAN bus protocol for IoT applications. *The 13th IEEE International Conference on Computer Engineering and Systems*. Cairo, Egypt, pages 275-278, 2018.

[3] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3): 362-386, 2019.

[4] 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*, 16(2): 1417-1426, 2020.

[5] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84: 317-331, 2018.

[6] G. Chen, Y. Mao, and C. K. Chui. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons & Fractals*, 21(3): 749-761, 2004.

[7] Q. Liu, P. Li, M. Zhang, Y. Sui, and H. Yang. A novel image encryption algorithm based on chaos maps with Markov properties. *Communication Nonlinear Science and Numerical Simulation*, 20(2): 506-515, 2015.

[8] Z. Guan, F. Huang, and W. Guan. Chaos-based image encryption algorithm. *Physics Letters A*, 346(1-3): 153-157, 2005.

[9] S. Behnia, A. Akhshani, S. Ahadpour, H. Mahmodi, and A. Akhavan. A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps. *Physics Letters A*, 366(4-5): 391-396, 2007.

[10] 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*, 9(1): 212-220, 2001.

[11] A. Akhenak, M. Chadli, J. Ragot, and D. Maquin. Unknown input multiple observer based approach-Application to secure communications. *1st IFAC Conference on Analysis and Control of Chaotic Systems*, 39(8): 172-177, 2006.

[12] Z. A. Biron, S. Dey, and P. Pisu. Real-time detection and estimation of denial of service attack in connected vehicle systems. *IEEE Transactions on intelligent transportation systems*. 19(12): 3893-3902, 2018.

[13] M. Zhu and S. Martinez. On resilient consensus against replay attacks in operator-vehicle networks. *American Control Conference*, Fairmont Queen Elizabeth, Montréal, pages 3553-3558, 2012.

[14] N. Bezzo, J. Weimer, Y. Du, O. Sokolsky, S. H. Son, and I. Lee. A stochastic approach for attack resilient UAV motion planning. *American Control Conference*, Boston, MA, pages 1366-1372, 2016.

[15] N. Bezzo, J. Weimer, M. Pajic, O. Sokolsky, G. J. Pappas, and I. Lee. Attack resilient state estimation for autonomous robotic systems. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, pages 3692-3698, 2014.

[16] X. He, E. Hashemi, and K. H. Johansson. Distributed control under compromised measurements: Resilient estimation, attack detection, and vehicle platooning. *arXiv:2010.09661*, 2020.

[17] F. J. Bejarano and A. Pisano. Switched observers for switched linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, 56(3): 681-686, 2011.

[18] K. Lian, C. Chiu, T. Chiang and P. Liu. Secure communications of chaotic systems with robust performance via fuzzy observer-based design. *IEEE Transactions on Fuzzy Systems*, 9(1): 212-220, 2001.

[19] M. Hou and P. C. Müller. Design of observer for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, 37(6): 871-875, 1992.

[20] H. K. Khalil. *Nonlinear Systems*. Prentice hall, New Jersey, 2002.

[21] P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4(4): 421-448, 1994.

[22] R. A. Horn and C. R. Johnson. *Matrix Analysis Second Edition*. Cambridge University Press, Ney York, NY, USA, 2013.

[23] E. D. Sontag. Input to state stability: Basic concepts and results. *Nonlinear and Optimal Control Theory*, 1932: 163-220, 2008.

[24] B. Ingalls and E. D. Sontag. Generalizations of asymptotic gain characterizations of ISS to input-to-output stability. *Proceedings of the American Control Conference*, Arlington, VA, pages 2279-2284, 2001.

[25] S. Dean, N. Matni, B. Recht, and V. Ye. Robust guarantees for perception-based control. *Proceedings of Machine Learning Research*, 120: 1-10, 2020.

[26] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

[27] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, Edinburgh, pages 958-962, 2003.