# Perfectly Undetectable Reflection and Scaling False Data Injection Attacks via Affine Transformation on Mobile Robot Trajectory Tracking Control

Jun Ueda, *Senior Member, IEEE*, and Hyukbin Kwon, *Student Member, IEEE*

***Abstract*—With the increasing integration of cyber-physical systems (CPS) into critical applications, ensuring their resilience against cyberattacks is paramount. A particularly concerning threat is the vulnerability of CPS to deceptive attacks that degrade system performance while remaining undetected. This article investigates perfectly undetectable false data injection attacks (FDIAs) targeting the trajectory tracking control of a nonholonomic mobile robot. The proposed attack method utilizes affine transformations of intercepted signals, exploiting weaknesses inherent in the partially linear dynamic properties and symmetry of the nonlinear plant. The feasibility and potential impact of these attacks are validated through experiments using a Turtlebot 3 platform, highlighting the urgent need for sophisticated detection mechanisms and resilient control strategies to safeguard CPS against such threats. Furthermore, a novel approach for detection of these attacks called the state monitoring signature function (SMSF) is introduced. An example SMSF, a carefully designed function resilient to FDIA, is shown to be able to detect the presence of an FDIA through signatures based on system states.**

***Index Terms*—Affine transformation, false data injection attack (FDIA), mobile robots, nonholonomic constraints, nonlinear kinematics, security, stability, trajectory tracking.**

## I. Introduction

VIRTUALLY all current robotic systems are interconnected through computer networks for exchanging sensor measurements, control commands, and other information for monitoring and controlling purposes [1]. Mobile robots are examples of such systems and have become integral to a broad spectrum of applications, particularly in scenarios where human intervention is either impractical or inefficient. These applications range from industrial automation and logistics, where mobile robots handle materials, to exploration and data collection in hazardous environments, such as deep-sea locations, disaster sites, and space missions [2]. Given the increasing reliance on mobile robots for critical tasks and their operation in potentially unsecured or remote environments, ensuring the robustness of these systems against cyberattacks is an important area of research [3].

The operation of these mobile robots often relies on networked communication systems to receive commands and transmit data back to the operators or control servers. This networked nature, while enabling remote and autonomous operations, is also susceptible to cybersecurity threats. One significant threat is false data injection attacks (FDIAs) [3], [4], [5], where an attacker manipulates the data being sent to or from the robot, or both, leading to incorrect actions, decision-making based on false information, or even taking control of the robot's operations. For instance, in an FDIA, the data regarding the robot's location or sensor measurements could be compromised, misleading the navigation system and causing the robot to deviate from its intended task. FDIA can be applied to hinder the rate of operation, induce denial of service conditions, change the region of exploration, and even cause physical damage to the surroundings. In more sophisticated scenarios, as shown in Fig. 1, attackers could inject false data to make the robot's system believe it is operating normally, referred to as undetectable or stealthy FDIAs [6], [7], while it performs unintended tasks. Cyberattacks, such as FDIAs, can be particularly dangerous because the resources to launch such an attack can be acquired easily. In the analysis of cyberattack threats, focus should be set on attack capability rather than motives, as even less motivated actors can launch an attack [6], [8].

Stealthy and undetectable attacks are characterized by their increased difficulty for operators to detect. In stealthy attacks, an attacker capable of intercepting the original messages can inject the attack with partial or no knowledge of the plant, ensuring that the changes remain below the threshold of an attack detector [6]. In the case of an undetectable attack, the attacked signals coincide with those that are within the regular operating range, causing faults and standard detectors to fail [9]. Perfectly undetectable attacks are those where there is no change in observed states, even though the closed-loop system is under attack and performs unintended motions. Similar attacks to those introduced in this article have been discussed as covert attacks, as proposed in [10], whereby if the attacker has perfect knowledge of the plant, it is possible to mask the attack from the controller's perspective.

Most works on covert attacks address linear time invariant [4], [9], [10], [11] systems. FDIA has been implemented in systems with moderate nonlinearities [12]. In this case, a simplified

Received 18 March 2025; revised 1 August 2025; accepted 16 September 2025. Date of publication 28 October 2025; date of current version 14 November 2025. This work was supported by the National Science Foundation under Grant 2112793 and Grant 2516189. This article was recommended for publication by Associate Editor E. Montijano and Editor P. R. Giordano upon evaluation of the reviewers' comments. *(Jun Ueda and Hyukbin Kwon are co-first authors), (Corresponding author: Hyukbin Kwon.)*

The authors are with the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0405 USA (e-mail: jun.ueda@me.gatech.edu; bin.kwon@gatech.edu).

Digital Object Identifier 10.1109/TRO.2025.3626620

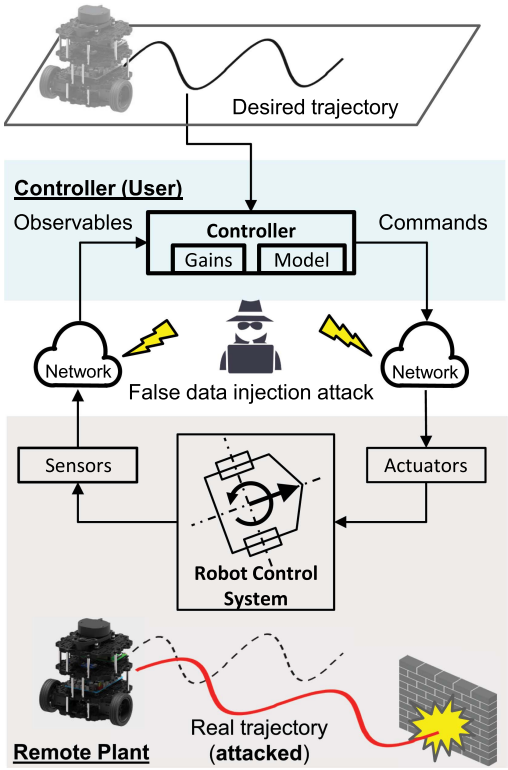



Fig. 1. Conceptual diagram of FDIA on remote mobile robot control system.

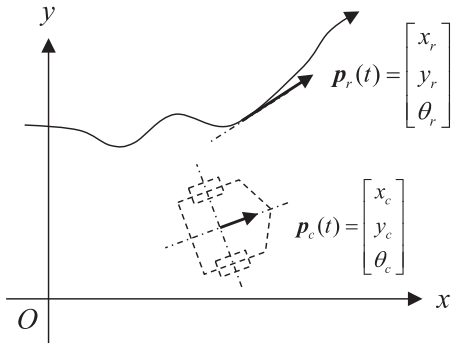



Fig. 2. Mobile robot desired and current postures. The same notations are used as in [26].

linearized version of the actual dynamics is used for the basis of the attack. Other considerations of nonstealthy FDIA to nonlinear systems have been made to a class of nonlinear systems [13]. In contrast, this article specifically discusses perfectly undetectable FDIA applied to nonlinear mobile robot dynamics.

The robustness of closed-loop systems to account for uncertainties, disturbances, and sensor noise is a well-established and extensively studied field of research. Common compensation strategies from the control-theoretical standpoint include robust optimal control [14], adaptive control [15], and state and disturbance compensation [16]. For anomaly detection associated with FDIA, a specific strategy involves a model-based control approach: the controller compares the observed plant behaviors induced by its control commands with those simulated based on a nominal plant dynamic model [6], [7], [9]. Any discrepancies identified through this process could indicate potential false-data injection, disturbances, or plant uncertainties. Encrypting communication lines or control algorithms [17], [18] adds another layer of protection. However, in a scenario where the attack already has access to the network, such as an insider attack or a data breach, conventional encryption alone does not guarantee security. Advanced encryption techniques, such as homomorphic encryption schemes, possess known malleability vulnerabilities that can be exploited to apply FDIA [11], [19], [20], [21], [22]. Conversely, if the controller observes small changes under the detection threshold (i.e., stealthy) or none at all (i.e., perfectly undetectable) in the plant dynamics between normal and attacked states, model-based anomaly detection would be ineffective [10], [23], [24], [25].

The significance of this article lies in the formulation of a generalized FDIA that involves coordinated multiplicative and additive data injections on both control commands and observables, taking form of affine transformations. Unlike covert attacks that require extensive computation and manipulation of the closed-loop system with complete knowledge of the plant dynamics [25], this relatively simplistic and static FDIA allows attackers to execute perfectly undetectable attacks on remotely controlled mobile robots. Employing a classical two-wheel mobile robot kinematic model as a case study, this article demonstrates how the inherent structure of commonly used nonlinear robot dynamics, from commands to outputs, enables a range of perfectly undetectable FDIAs. This vulnerability persists regardless of the type of trajectory control (e.g., [26]), resulting in undetected failures within the controller's attack detection algorithms.

As a countermeasure, this article proposes a state monitoring signature function (SMSF) approach along with an associated implementation architecture to continuously monitor for indications of perfectly undetectable FDIAs. A signature function can be constructed from polynomial functions that are resilient to scaling and reflection attacks. While not permanently secure, the signature function can be designed to be difficult for the attacker to adversarially estimate for spoofing attacks. SMSF possesses useful properties for attack detection over previously suggested methods. The SMSF approach differs from hash functions [27] in that it operates on continuous and dynamic system states rather than static data. This allows for verification of integrity based on comparison of plant–controller output of the SMSF. In addition, SMSF can be implemented as a full software solution, avoiding the need for an additional dynamic component often required by auxiliary systems [25].

The rest of this article is organized as follows. Section II provides preliminaries on well-known mobile robot dynamics and representative trajectory control methods. Section III discusses perfectly undetectable FDIA on nonlinear control systems and introduces affine transformation-based formulations. Section IV offers two solutions to the perfectly undetectable FDIA problem. Section IV-C analyzes the stability of the closed-loop system under perfectly undetectable FDIA. Section V presents experimental results. Section VI introduces an SMSF as a countermeasure to perfectly undetectable FDIAs. While promising, this method

has its own limitations. Section VII discusses key observations and limitations. Finally, Section VIII concludes this article.

## II. Preliminaries on Mobile Robot Dynamic and Control

Well-known dynamic equations of a typical two-wheel mobile robot on a 2-D plane (e.g., [26]) are given in the following for readers' convenience. Interested readers can find extensive resources online and in the literature [28].

The position of the robot can be represented with three degrees of freedom (DOF), as shown in Fig. 2, where $x_c$ and $y_c$ are positions and $\theta_c$ is the orientation in the global frame

$$\boldsymbol{p}_c = \begin{bmatrix} x_c \\ y_c \\ \theta_c \end{bmatrix}. \tag{1}$$

The mobile robot can only be moved in two DOFs due to its nonholonomic constraints

$$\boldsymbol{q} = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2}$$

where $v$ and $\omega$ are the linear and angular velocities in the robot's local coordinate frame, respectively.

A typical controller for tracking a given reference trajectory $\boldsymbol{r}(t) \in \mathcal{R}^{2\times 2}$ is implemented. The inputs to the controller are the reference posture $\boldsymbol{p}_r = [x_r, y_r, \theta_r]^T$ and the robot's current posture $\boldsymbol{p}_c$. Typically, both the error between the reference posture and current posture $\boldsymbol{p}_e = \mathbf{R}_{\text{rot}Z}(\theta_c)(\boldsymbol{p}_r - \boldsymbol{p}_c)$ where $\mathbf{R}_{\text{rot}Z}(\theta_c)$ is the rotation of $\theta_c$ about the Z axis, and the reference linear and angular velocities $\boldsymbol{q}_r = [v_r, \omega_r]^T$ computed from $\boldsymbol{r}(t)$ are used.

The dynamics of the mobile robot are given in the following equation as a first-order nonlinear equation:

$$\dot{\boldsymbol{p}}_c = \boldsymbol{J}(\boldsymbol{p}_c)\boldsymbol{q} \tag{3}$$

where $\boldsymbol{J}$ is the Jacobian matrix that maps the control command $\boldsymbol{q}$ onto the time derivative of $\boldsymbol{p}_c$

$$\boldsymbol{J} = \begin{bmatrix} \cos\theta_c & 0 \\ \sin\theta_c & 0 \\ 0 & 1 \end{bmatrix}. \tag{4}$$

The controller outputs the input vector $\boldsymbol{q}$ as a control command that is sent via the communication channel.

One of the well-cited tracking control schemes was proposed by Kanayama et al. [26], which this article adopts as a representative control scheme

$$\boldsymbol{q} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos\theta_e + k_x x_e \\ \omega_r + v_r(k_y y_e + k_\theta \sin\theta_e) \end{bmatrix} \tag{5}$$

where the error $\boldsymbol{q}_e = 0$ was proven to be globally asymptotically stable with a Lyapunov function defined as: $V = \frac{1}{2}({x_e}^2 + {y_e}^2) + (1 - \cos\theta_e)/k_y$. It should be noted that the attacker is not required to know the tracking control type or its gains to successfully implement a perfectly undetectable FDIA presented in this article.

## III. Perfectly Undetectable FDIA: Coordinating Attacks on Observables and Control Commands

### A. Fundamental Equations

Consider a general nonlinear dynamic plant in affine form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{6}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{7}$$

where $\boldsymbol{f}$ and $\boldsymbol{g}$ are both Lipschitz continuous functions, and a state feedback law is given by

$$\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{x}). \tag{8}$$

It is assumed that a remote dynamic plant, described by (6) and (7), is controlled by a centralized controller defined in (8). A generalized form of FDIA that involves coordinated multiplicative and additive data injections into both control commands and observables is depicted in Fig. 3(a)

$$\tilde{\boldsymbol{x}} = \boldsymbol{\alpha}(\boldsymbol{x}) \tag{9}$$

where $\boldsymbol{\alpha}$ is a static observables attack function, and its inverse function $\boldsymbol{\alpha}^{-1}$ exists. Similarly to the control command, $\tilde{\boldsymbol{u}}$ is a compromised (attacked) control command vector resulting from the attack by a static attack function, $\boldsymbol{\beta}$

$$\tilde{\boldsymbol{u}} = \boldsymbol{\beta}(\boldsymbol{u}). \tag{10}$$

Under the attack defined by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the controller perceives the plant dynamics based on the command and the compromised observables, i.e.,

$$\begin{aligned} \dot{\tilde{\boldsymbol{x}}} &= \frac{\partial \boldsymbol{\alpha}(\boldsymbol{x})}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} \\ &= \frac{\partial \boldsymbol{\alpha}(\boldsymbol{x})}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{\beta}(\boldsymbol{u}) \\ &= \left.\frac{\partial \boldsymbol{\alpha}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{\alpha}^{-1}(\tilde{\boldsymbol{x}})} \boldsymbol{f}(\boldsymbol{\alpha}^{-1}(\tilde{\boldsymbol{x}})) + \boldsymbol{g}((\boldsymbol{\alpha}^{-1}(\tilde{\boldsymbol{x}}))\boldsymbol{\beta}(\boldsymbol{u}). \end{aligned} \tag{11}$$

Following the nominal plant dynamics (6) and (7), let us define $\boldsymbol{x}'$ and $\boldsymbol{y}'$ that evolve with the same input $\boldsymbol{u}$ introduced by the attacker to mislead the controller into believing that the plant is operated normally:

$$\dot{\boldsymbol{x}}' = \boldsymbol{f}(\boldsymbol{x}') + \boldsymbol{g}(\boldsymbol{x}')\boldsymbol{u} \tag{12}$$

$$\boldsymbol{y}' = \boldsymbol{h}(\boldsymbol{x}') \tag{13}$$

$$\boldsymbol{x}'(0) = \boldsymbol{x}(0). \tag{14}$$

*Proposition 1 (Indistinguishable plant responses amidst perfectly undetectable FDIA):* If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ exist such that the following conditions hold, then a perfectly undetectable FDIA is achieved where $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}'(t) \forall t \geq 0$, *regardless* of the controller $\boldsymbol{k}(\boldsymbol{x})$.

1) *Condition 1 (observing the nominal initial conditions):* $\boldsymbol{x}(0) = \boldsymbol{\alpha}(\boldsymbol{x}(0))$ ensures that the observed state of the mobile robot at the start of the attack matches its actual state. This condition is crucial because if the attacker modifies the initial observed state, then the controller

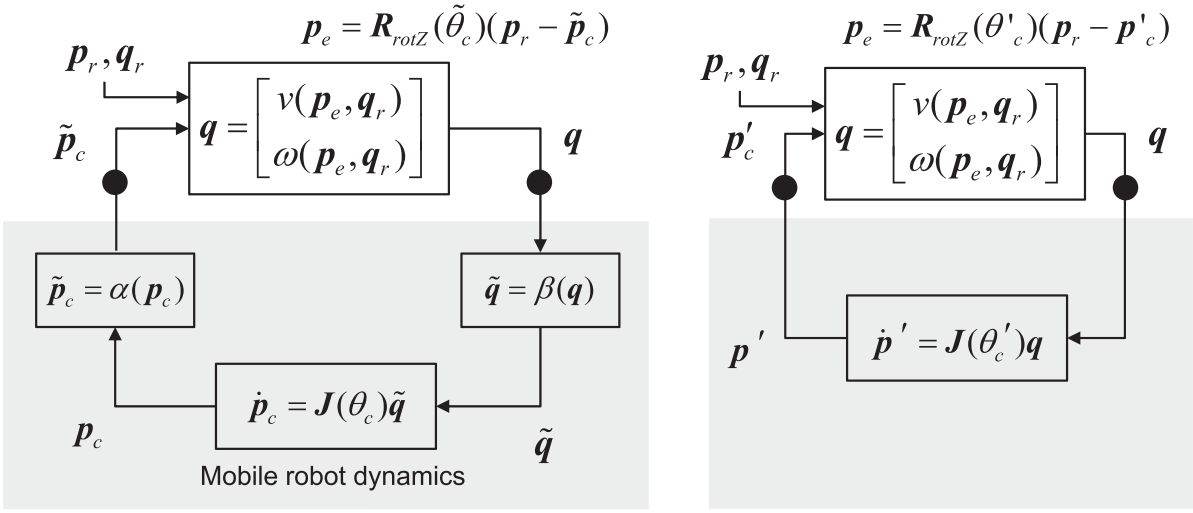



Fig. 3. Perfectly undetectable FDIA on remote mobile robot control system based on affine transformations. (a) Attacked control system with coordinated FDIA on the commands and observables. (b) Plant dynamics as perceived by the controller, indistinguishable from the nominal robot behavior and thus undetectable.

would immediately detect a discrepancy and recognize the presence of an attack. In essence, the attack must begin by presenting the controller with the true initial state of the robot.

2) *Condition 2 (observing the nominal dynamics):*

$$\left.\frac{\partial \boldsymbol{\alpha}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{\alpha}^{-1}(\tilde{\boldsymbol{x}})} \boldsymbol{f}(\boldsymbol{\alpha}^{-1}(\tilde{\boldsymbol{x}})) + \boldsymbol{g}((\boldsymbol{\alpha}^{-1}(\tilde{\boldsymbol{x}}))\boldsymbol{\beta}(\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}') + \boldsymbol{g}(\boldsymbol{x}')\boldsymbol{u} \quad \forall \boldsymbol{u}. \tag{15}$$

Equation (15) ensures that the compromised system's dynamics, as observed by the controller, is identical to the nominal (unattacked) system's dynamics for all possible inputs, $\boldsymbol{u}$.

*Proof:* This proposition is a direct corollary of the Picard–Lindelöf theorem [29], or the uniqueness of the solution to an initial value problem for an ordinary differential equation. The first condition must be satisfied for the controller to observe the same initial state: $\tilde{\boldsymbol{x}}(0) = \boldsymbol{\alpha}(\boldsymbol{x}(0)) = \boldsymbol{x}'(0) = \boldsymbol{x}(0)$; otherwise, an attack detector in the controller would immediately detect a data falsification. Once the first condition is satisfied, $\tilde{\boldsymbol{x}}$, which evolves according to (11), and $\tilde{\boldsymbol{x}}'$, which evolves according to (12), yield identical values at all times when the same $\boldsymbol{u}$ is applied. ■

*Remark 1 (Imperceptibility from the controller perspective):* Let $x(t, x(0), k(x), \alpha, \beta)$ denote the state of a control affine plant (6) with initial condition $x(0)$, state feedback controller $k(x)$, and FDIA functions $\alpha(x), \beta(u)$. Also, let $\tilde{x}(t, x(0), k(x), \alpha, \beta)$ denote the state measures that the controller (user) receives from the attacked target plant. The attack is perfectly undetectable from the controller's perspective if $x(t, x(0), k(x), 1, 1) = \tilde{x}(t, x(0), k(x), \alpha, \beta), t \geq 0$, because the controller receives state measurements identical to those it would receive if there were no attack, even though the actual system state is different. Appendix A discusses conditions from the plant perspective.

### B. Specific Conditions of Perfectly Undetectable FDIA on Mobile Robot Control

The mobile robot dynamic equation (3) is a special case of (6) and (7) where

$$\boldsymbol{x} = \boldsymbol{p}_c \tag{16}$$

$$\boldsymbol{u} = \tilde{\boldsymbol{q}} \tag{17}$$

$$f(\boldsymbol{x}) = \boldsymbol{0} \tag{18}$$

$$g(\boldsymbol{x}) = \boldsymbol{J}(\theta_c) = \begin{bmatrix} \cos\theta_c & 0 \\ \sin\theta_c & 0 \\ 0 & 1 \end{bmatrix} \tag{19}$$

$$h(\boldsymbol{x}) = \boldsymbol{x} = \boldsymbol{p}_c. \tag{20}$$

The attack functions are also defined as

$$\tilde{\boldsymbol{p}}_c = \boldsymbol{\alpha}(\boldsymbol{p}_c) \tag{21}$$

$$\tilde{\boldsymbol{q}} = \boldsymbol{\beta}(\boldsymbol{q}). \tag{22}$$

The dynamic relationship between the command $\boldsymbol{q}$ and the observable $\tilde{\boldsymbol{p}}_c$ illustrated as a shaded region in Fig. 3(a) is given as

$$\dot{\tilde{\boldsymbol{p}}}_c = \frac{\partial \boldsymbol{\alpha}(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c}\dot{\boldsymbol{p}}_c = \frac{\partial \boldsymbol{\alpha}(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c}\boldsymbol{J}(\theta_c)\tilde{\boldsymbol{q}} = \frac{\partial \boldsymbol{\alpha}(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c}\boldsymbol{J}(\theta_c)\boldsymbol{\beta}(\boldsymbol{q}). \tag{23}$$

When the controller perceives the attacked plant dynamics as matching the nominal plant dynamics, a perfectly undetectable FDIA is considered to be achieved, as illustrated in Fig. 3(b), i.e.,

$$\dot{\tilde{\boldsymbol{p}}}_c = \frac{\partial h(\boldsymbol{p}')}{\partial \boldsymbol{p}'}\dot{\boldsymbol{p}}' = \boldsymbol{J}(\theta'_c)\boldsymbol{q} \tag{24}$$

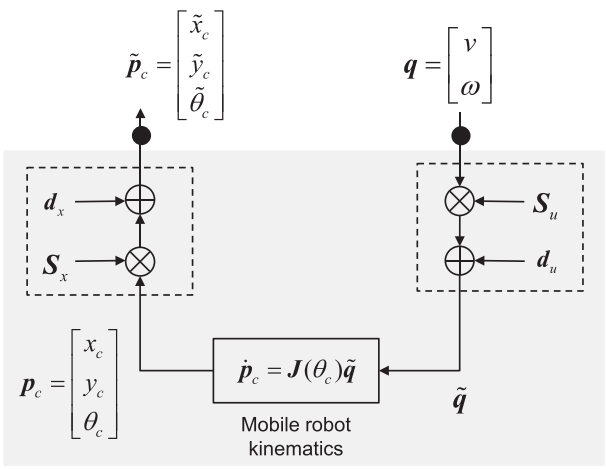



Fig. 4. Affine transformation-based coordinated FDIA on mobile robot control system.

where $p' = [x'_c, y'_c, \theta'_c]$ is a *fake* state variable vector. In fact, $p' \neq p$, and the expected behavior is different from the actual behavior. If the controller observes

$$p' = \tilde{p}_c = \alpha(p_c) \tag{25}$$

from (24) and (25), then the observed dynamics by the controller becomes equivalent to

$$\dot{p}' = J(\theta'_c)q \tag{26}$$

which matches with the nominal dynamics (3), achieving a perfectly undetectable FDIA, *regardless* of the controller that generates $q$. In a later section, a theorem with conditions, including one about the initial conditions, will be given.

### C. Problem Formulation Using Affine Transformation-Based FDIA

The main objective of this article is to discuss the existence of attack functions $\alpha$ and $\beta$ for (3) that yield (26). For simplicity, let us assume that the attacker opts for a transformation in an affine form, as shown in Fig. 4, instead of general nonlinear attack functions. This assumption is not entirely unrealistic. When the communication lines are encrypted using homomorphic encryption algorithms, they impose limited computational capabilities on the attacker [11], allowing only simple operations, such as multiplication and addition, to be performed on the original messages in the communication lines. This type of vulnerability is known as a malleability attack [19], [20].

The attack to the observables in an affine form is given as follows:

$$\tilde{p}_c = \alpha(p_c) = S_x p_c + d_x \tag{27}$$

or, alternatively, in the form of homogeneous transformation

$$\begin{bmatrix} \tilde{p}_c \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & d_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_c \\ 1 \end{bmatrix} \tag{28}$$

where $S_x \in \mathbb{R}^{n\times n}$ represents an arbitrary transformation, such as scaling, shear, and rotation, and $d_x \in \mathbb{R}^n$ represents a translation introducing an offset. This article assumes that $S_x$ and $d_x$ are constants.

Similarly, as with the control command, $S_u \in \mathbb{R}^{m\times m}$ represents an arbitrary transformation, and $d_u \in \mathbb{R}^m$ represents a translation introducing an offset. This article assumes that $S_u$ and $d_u$ are constants

$$\tilde{q} = \beta(q) = S_u q + d_u \tag{29}$$

$$\begin{bmatrix} \tilde{q} \\ 1 \end{bmatrix} = \begin{bmatrix} S_u & d_u \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q \\ 1 \end{bmatrix}. \tag{30}$$

In the literature, most studies considered either additive or multiplicative FDIA on control commands or observables. For example, Zhu et al. [30] studied both multiplicative and additive data injections, assuming that the observables remained uncompromised. Representing FDIAs in the affine form with (27) and (29) allows for more generalized analyses. It should be mentioned that simultaneous FDIA on the commands and observables is not necessarily a new concept. Past works on covert attacks introduced a similar structure in which the attacker implements an additional dynamic controller between the commands and observables [25]. In contrast, this article formulates perfectly undetectable FDIAs in terms of affine transformations, representing, to the best of the authors' knowledge, for the first time, this has been done on nonlinear robot system dynamics.

Based on the aforementioned analysis and *Proposition 1*, the perfectly undetectable FDIA problem, which is specific to the mobile robot dynamics, can be defined as follows.

*Definition 1 (Perfectly undetectable FDIA problem on mobile robot dynamics):* For the nominal plant dynamic equation (3) with the Jacobian matrix (4), if $S_x$, $d_x$, $S_u$, and $d_u$ exist such that "fake" state variables $p'$ can be defined and the following conditions hold, then a perfectly undetectable FDIA is implemented.

1) *Condition 1 (Same initial condition):* $\tilde{p}_c(0) = S_x p_c(0) + d_x = p_c(0)$.
2) *Condition 2 (Same observed dynamics):* The observed plant dynamics by the controller $\dot{\tilde{p}}_c = S_x J(\theta_c)(S_u q + d_u) = S_x \dot{p}_c$ is equivalent to the nominal dynamics $\dot{p}' = J(\theta'_c)q$ evolved by the same command $q(t)$, where the attacked dynamics is given by $\dot{p}_c = J(\theta_c)(S_u q + d_u)$ and $\theta_c = [0\ 0\ 1]p_c$. In short, $J(\theta_c)(S_u q + d_u) = J(\theta'_c)q$ must be satisfied.

Specific solutions are provided in the next section.

## IV. Derivation of Attack Parameters for Perfectly Undetectable FDIA

### A. Attackability Analysis of Mobile Robot Jacobian Matrix

Equation (4) reveals a block-diagonal structure, with its (3, 1) element being constant and decoupled from the (1, 1) and (2, 1) elements. This indicates that the robot's angle $\theta_c$ is governed by a first-order linear equation that solely depends on the input $\omega$.

This section first considers the structure of $\boldsymbol{S}_u$. Assuming that $\boldsymbol{S}_u = \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}$, the following proposition is obtained.

*Proposition 2 (General form of $\boldsymbol{S}_u$ and associated requirements):* The general form of $\boldsymbol{S}_u$ is given as a diagonal form, $\boldsymbol{S}_u = \begin{bmatrix} \beta_{11} & 0 \\ 0 & \pm 1 \end{bmatrix}$ subject to the following requirements.

1) $\beta_{12} = 0$ and $\beta_{21} = 0$ under the assumption that $\boldsymbol{d}_x$ is a constant.
2) $\beta_{22} = \pm 1$.

*Proof:* Consider the evolution of $\tilde{\boldsymbol{p}}_c = \int_0^t J(\theta_c)\boldsymbol{S}_u \boldsymbol{q}\, \mathrm{d}t$, i.e.,

$$x_c = \beta_{11}\int_0^t v\cos(\theta_c)\,\mathrm{d}t + \beta_{12}\int_0^t \omega\cos(\theta_c)\,\mathrm{d}t = \beta_{11}\tilde{x}_c - \beta_{12}\sin(\theta_c) \tag{31}$$

$$y_c = \beta_{11}\tilde{y}_c + \beta_{12}\cos(\theta_c) \tag{32}$$

$$\theta_c = \beta_{21}\int_0^t v\,\mathrm{d}t + \beta_{22}\tilde{\theta}_c. \tag{33}$$

The second term in both (31) and (32) is a nonlinear function of $\theta_c$ and thus time-dependent. Unless $\boldsymbol{d}_x$ is computed from $\theta_c$ in real time, the attacker cannot eliminate this term to realize a perfectly undetectable FDIA. Similarly, the second term in (33) is the length of the path produced by the robot. $\boldsymbol{p}_c$ does not store such information. Unless $\boldsymbol{d}_x$ includes an integration of $v$ over time, the attacker cannot eliminate this term to realize a perfectly undetectable FDIA. These observations contradict the assumption of an affine transformation, leading to $\beta_{12} = \beta_{21} = 0$. Regarding (33), since $\beta_{21} = 0$, $\tilde{\theta}_c = \beta_{22}\tilde{\theta}_c$. See Appendix B for Proposition B1 about the vulnerability of trigonometric functions. Considering $\cos(\tilde{\theta}_c) = \cos(\beta_{22}\theta_c)$, only $\beta_{22} = \pm 1$ is feasible. ■

*Remark 2 (Possible FDIA scenarios):* $\beta_{11}$ is a scaling factor that represents an attack on the linear velocity, termed a scaling attack. No attack is imposed on the linear velocity when $\beta_{11} = 1$. Also, $\beta_{11} = 0$ cannot be chosen since such an attack would be immediately detected by the controller; thus, $\beta_{11} \neq 0$. Since no attack is imposed on the angular velocity command when $\beta_{22} = 1$ (i.e., the trivial case), the only effective selection of $\beta_{22} = -1$, a scenario termed a reflection attack.

*Remark 3 (Future time-variant $\boldsymbol{d}_x$):* $\beta_{12} \neq 0$ and $\beta_{21} \neq 0$ may be used when $\boldsymbol{d}_x$ is time-variant. This consideration is beyond the scope of this particular article and will be addressed in future work.

### B. Main Result: Perfectly Undetectable FDIA Solutions

Based on the aforementioned analysis, the following theorem is obtained that shows the existence of specific solutions for affine transformation-based perfectly undetectable FDIAs.

*Theorem 1 (Specific FDIA solutions to mobile robot dynamics (see: Proposition 1 )*

1) *Condition 1*: $(\boldsymbol{I}_2 - \boldsymbol{S}_x)\boldsymbol{p}_c(0) = \boldsymbol{d}_x$
2) *Condition 2*: $\boldsymbol{d}_u = 0$ and
   a) *Condition 2-1:* Reflection attack

$$\boldsymbol{S}_x = \begin{bmatrix} \frac{1}{\beta_{11}}\cos(2\theta_c(0)) & \frac{1}{\beta_{11}}\sin(2\theta_c(0)) & 0 \\ \frac{1}{\beta_{11}}\sin(2\theta_c(0)) & -\frac{1}{\beta_{11}}\cos(2\theta_c(0)) & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{34}$$

   and $\boldsymbol{S}_u = \begin{bmatrix} \beta_{11} & 0 \\ 0 & -1 \end{bmatrix}$, where $\beta_{11} \neq 0$ (see *Remark 2*), termed a *reflection* attack.
   b) *Condition 2-2:* Scaling attack

$$\boldsymbol{S}_x = \begin{bmatrix} \frac{1}{\beta_{11}} & 0 & 0 \\ 0 & \frac{1}{\beta_{11}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{35}$$

   and $\boldsymbol{S}_u = \begin{bmatrix} \beta_{11} & 0 \\ 0 & 1 \end{bmatrix}$, termed a *scaling* attack.

*Proof:* Note that the observation at $t = 0$ must be unchanged, i.e., $\boldsymbol{p}_c(0) = \boldsymbol{S}_x\boldsymbol{p}_c(0) + \boldsymbol{d}_x$, Condition 1

$$(\boldsymbol{I}_2 - \boldsymbol{S}_x)\boldsymbol{p}_c(0) = \boldsymbol{d}_x \tag{36}$$

is obtained.

A possible attack may be a reflection attack on the angular velocity, i.e., $\boldsymbol{S}_u = \begin{bmatrix} \beta_{11} & 0 \\ 0 & \pm 1 \end{bmatrix}$,

$$\boldsymbol{\beta}(\boldsymbol{q}) = \boldsymbol{S}_u\boldsymbol{q} + \boldsymbol{d}_u = \begin{bmatrix} \beta_{11} & 0 \\ 0 & \pm 1 \end{bmatrix}\begin{bmatrix} v \\ \omega \end{bmatrix} + \boldsymbol{d}_u = \begin{bmatrix} \beta_{11} v \\ \pm\omega \end{bmatrix} + \boldsymbol{d}_u \tag{37}$$

yielding

$$\dot{\tilde{\boldsymbol{p}}}_c = \frac{\partial \boldsymbol{\alpha}(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c}\boldsymbol{J}(\theta_c)\boldsymbol{\beta}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\alpha}(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c}\boldsymbol{J}(\theta_c)(\boldsymbol{S}_u\boldsymbol{q} + \boldsymbol{d}_u) = \frac{\partial \alpha(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c}\begin{bmatrix} \beta_{11}\cos\theta_c & 0 \\ \beta_{11}\sin\theta_c & 0 \\ 0 & \pm 1 \end{bmatrix}\boldsymbol{q} + \frac{\partial \boldsymbol{\alpha}(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c}\boldsymbol{J}(\theta_c)\boldsymbol{d}_u. \tag{38}$$

Note that the second term that works as a bias must be $\frac{\partial \boldsymbol{\alpha}(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c}\boldsymbol{J}(\theta_c)\boldsymbol{d}_u = 0$ to observe the nominal dynamics by the controller. Since $\boldsymbol{J}(\theta_c)$ is state-dependent and time-variant, the attacker must choose $\boldsymbol{d}_u = \boldsymbol{0}$.

If $\boldsymbol{J}(\theta_c') = \frac{\partial \boldsymbol{\alpha}(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c}\begin{bmatrix} \beta_{11}\cos\theta_c & 0 \\ \beta_{11}\sin\theta_c & 0 \\ 0 & \pm 1 \end{bmatrix}$, then the perfectly undetectable FDIA is successfully implemented. Consider when $\beta_{22} = -1$ (reflection attack). Since $\frac{\partial \boldsymbol{\alpha}(\boldsymbol{p}_c)}{\partial \boldsymbol{p}_c} = \boldsymbol{S}_x$

$$\boldsymbol{J}(\theta_c') = \begin{bmatrix} \cos\theta_c' & 0 \\ \sin\theta_c' & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{S}_x\begin{bmatrix} \beta_{11}\cos\theta_c & 0 \\ \beta_{11}\sin\theta_c & 0 \\ 0 & -1 \end{bmatrix}$$

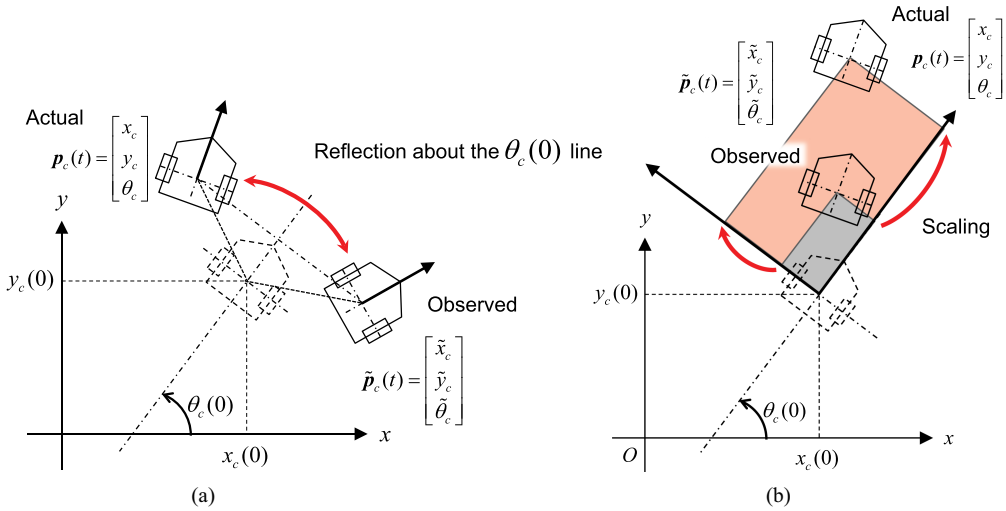



Fig. 5. Perfectly undetectable FDIA solutions: (a) reflection and (b) scaling attacks.

$$= \left[\begin{array}{cc|c} \frac{1}{\beta_{11}}\boldsymbol{R}_{\text{ref}}(\theta_c \to \theta'_c) & & \begin{matrix} 0 \\ 0 \end{matrix} \\ \hline 0 \quad 0 & & -1 \end{array}\right] \begin{bmatrix} \beta_{11}\cos\theta_c & 0 \\ \beta_{11}\sin\theta_c & 0 \\ 0 & -1 \end{bmatrix}$$

$$= \left[\begin{array}{c|c} \frac{1}{\beta_{11}}\boldsymbol{R}_{\text{ref}}(\theta_c \to \theta'_c) \begin{bmatrix} \beta_{11}\cos\theta_c \\ \beta_{11}\sin\theta_c \end{bmatrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \hline 0 \quad 0 & 1 \end{array}\right] \tag{39}$$

where $\boldsymbol{R}_{\text{ref}}(\theta_c \to \theta'_c)$ is a reflection matrix to reflect $\theta_c$ to $\theta'_c = \tilde{\theta}_c$ about the $\theta_c(0)$ line, as in Fig. 5(a), given as follows:

$$\begin{aligned} \cos\theta'_c &= \cos(-\theta_c + 2\theta_c(0)) \\ &= \cos(\theta_c)\cos(2\theta_c(0)) + \sin(\theta_c)\sin(2\theta_c(0)) \end{aligned} \tag{40}$$

$$\begin{aligned} \sin\theta'_c &= \sin(-\theta_c + 2\theta_c(0)) \\ &= -\sin(\theta_c)\cos(2\theta_c(0)) + \cos(\theta_c)\sin(2\theta_c(0)) \end{aligned} \tag{41}$$

$$\begin{bmatrix} \cos\theta'_c \\ \sin\theta'_c \end{bmatrix} = \boldsymbol{R}_{\text{ref}}(\theta_c \to \theta'_c) \begin{bmatrix} \cos\theta_c \\ \sin\theta_c \end{bmatrix} \tag{42}$$

$$\therefore \boldsymbol{R}_{\text{ref}}(\theta_c \to \theta'_c) = \begin{bmatrix} \cos(2\theta_c(0)) & \sin(2\theta_c(0)) \\ \sin(2\theta_c(0)) & -\cos(2\theta_c(0)) \end{bmatrix} \tag{43}$$

yielding

$$\boldsymbol{S}_x = \begin{bmatrix} \frac{1}{\beta_{11}}\cos(2\theta_c(0)) & \frac{1}{\beta_{11}}\sin(2\theta_c(0)) & 0 \\ \frac{1}{\beta_{11}}\sin(2\theta_c(0)) & -\frac{1}{\beta_{11}}\cos(2\theta_c(0)) & 0 \\ 0 & 0 & -1 \end{bmatrix}. \tag{44}$$

Similarly, when $\beta_{22} = 1$, $\boldsymbol{S}_u$ only imposes a scaling attack without reflection, as illustrated in Fig. 5(b), i.e.,

$$\boldsymbol{S}_x = \begin{bmatrix} \frac{1}{\beta_{11}} & 0 & 0 \\ 0 & \frac{1}{\beta_{11}} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{45}$$

■

*Remark 4 (Initial conditions $\boldsymbol{p}_c(0)$ required by the attacker):* $\boldsymbol{p}_c(0)$ must be known by the attacker to satisfy Condition 1 at the onset of the attack. In the particular system used, the initial state is observable to the attacker as well. In the general case, if the initial conditions cannot be obtained, then the attacker can employ tools, such as state estimation, to meet condition 1. Time-variant attack parameters may be implemented to relax this condition, as previously mentioned in *Remark 3*. If the condition is not met, small deviations can be used to detect the attack until it settles, as explored in the authors' previous work [11].

*Remark 5 (Necessity of $\boldsymbol{d}_u = \boldsymbol{0}$):* Additive FDIA on control commands, $\boldsymbol{d}_u \neq \boldsymbol{0}$, is detectable and thus relatively easily compensated for by using traditional robust control methods, such as disturbance observers. This is required primarily due to the inertial property of the robot dynamics without an explicit static equilibrium shown in (18). Conversely, for other dynamic systems with a nonzero drift vector field term, $f(\boldsymbol{x}) \neq \boldsymbol{0}$, a nonzero $\boldsymbol{d}_u$ may need to be determined.

### C. Stability of the Closed-Loop System With Perfectly Undetectable FDIAs

Recall that *Proposition 1* indicates that the attacked system will remain convergent as long as a perfectly undetectable FDIA is implemented under a stabilizing controller, such as (5), *regardless* of the specific controller used. Nevertheless, this section provides a sketch of proof to confirm the stability of the attacked system for a specific control scheme.

*Proposition 3 (Stability of trajectory tracking control (Kanayama et al. [26] modified)):* For the control scheme that uses the compromised observables $\tilde{x}_e, \tilde{y}_e$ and $\tilde{\theta}_e$ due to FDIA

$$\boldsymbol{q} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r\cos\tilde{\theta}_e + k_x\tilde{x}_e \\ \omega_r + v_r(k_y\tilde{y}_e + k_\theta\sin\tilde{\theta}_e) \end{bmatrix} \tag{46}$$

$\tilde{\boldsymbol{q}}_e = 0$ is a stable equilibrium for the reference velocity $v_r > 0$.

*Sketch of proof:* Since a perfectly undetectable FDIA is implemented, $\dot{\tilde{\boldsymbol{p}}}_c = \boldsymbol{S}_x\boldsymbol{J}(\theta_c)\boldsymbol{S}_u\boldsymbol{q} = \boldsymbol{J}(\theta')\boldsymbol{q}$ holds. $\tilde{\boldsymbol{p}}_c$ evolves in exactly the same way as $\boldsymbol{p}'$ does. Therefore, the change of variables can be performed for the Jacobian matrix, i.e., $\dot{\tilde{\boldsymbol{p}}}_c = \boldsymbol{J}(\tilde{\theta}_c)\boldsymbol{q}$. Consequently, the error dynamics associated with the control scheme can be fully expressed in terms of $\tilde{x}_e, \tilde{y}_e,$ and $\tilde{\theta}_e$.

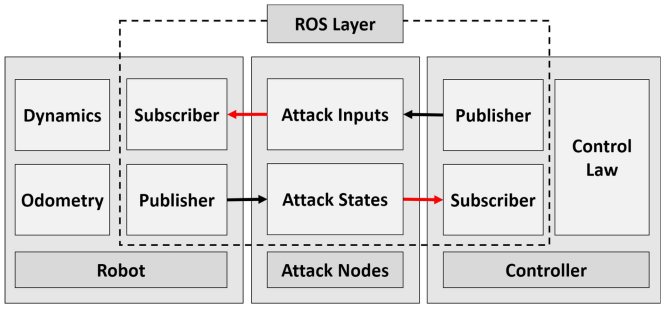



Fig. 6. Implemented ROS nodes for experimentation. Red arrows denote modified data.

Likewise, for a Lyapunov function candidate defined as: $\tilde{V} = \frac{1}{2}(\tilde{x}_e^2 + \tilde{y}_e^2) + (1 - \cos\tilde{\theta}_e)/k_y$, its time derivative can be expressed by

$$\dot{\tilde{V}} = -k_x\tilde{x}_e^2 - v_r k_\theta \sin^2\tilde{\theta}/k_y \leq 0 \tag{47}$$

resulting in the same conclusion shown in [26]. Since $\tilde{\boldsymbol{p}}_e(0) = \boldsymbol{p}_e{}'(0)$, the error dynamics between the observed plant and that of the nominal plant match exactly, confirming a perfectly undetectable FDIA. ■

## V. Experiments

### A. Mobile Robot Experimental Setup

A nonholonomic mobile robot (Turtlebot 3) with an onboard computer (Raspberry Pi 3) and a separate computer running Ubuntu Linux (11th Gen Intel Core i7-1165G7) functioning as the controller was used. Communication between the robot and the remote controller was established using TCP/IP with ROS 2. The design of the ROS network is shown in Fig. 6. Each attacker node modifies the published inputs and observables according to the preloaded attack scenario. Modified data shown in red arrows are received by the robot and the controller, respectively. Plant and controller nodes subscribe to the modified messages for use in the control loop. The computer runs the controller and attacker nodes, while an onboard single-board computer on the robot listens to the input commands and broadcasts its current state. The robot used Google Cartographer [31] for localization during the task. The controller node implements the controller presented in (46) [26] with gains $K_x = 2$, $K_y = 2000$, and $K_\theta = 100$. The controller was evaluated at 100 Hz, while errors and control inputs were logged at 50 Hz.

In order to satisfy Condition 1, the attacker was assumed to have knowledge of the robot's initial conditions. Because the controller also knows the initial condition of the system, incorrect application of FDIA to the initial conditions would lead to detection. The attacker is able to find the constant attack parameters to avoid detection by using their knowledge of the initial conditions. The attacker was also assumed to have knowledge of the structure of the specific Jacobian matrix $\boldsymbol{J}$ used for the operation of the robot. Note that if the structure of $\boldsymbol{J}$ is permuted, then the attack matrices should be permuted accordingly. On the other hand, the attacker was not required to know the geometries of the mobile robot or its mechanical details, such as wheel size, tread length, inertia, chassis material, and center of gravity. Furthermore, the attacker was not required to know any details implemented in the controller, such as the tracking controller type or gains.

In the experiments, the FDIA's computational load was negligible compared to that of the controller and could be applied without affecting the real-time control capability. For this experiment, the mobile robot was connected to the computer via an Ethernet cable to minimize time delay.

### B. Attack Scenarios and Results

Two attack scenarios are shown as illustrative examples of how proposed FDIA can affect actual systems, such as mobile robots. The attack parameters $\boldsymbol{S}_u$, $\boldsymbol{S}_x$, and $\boldsymbol{d}_x$ are determined as presented in Section IV-B. The reflection attack (see Scenarios 1 and 3) and scaling attack (see Scenario 2) are implemented with the attack matrices. $\boldsymbol{d}_x$ is chosen to be $(\boldsymbol{I}_3 - \boldsymbol{S}_x)\boldsymbol{p}_c(0)$ according to Condition 1 presented in Proposition 1. The initial conditions are set to be $\boldsymbol{p}_c(0) = [0, 0.02, 0]^T$ with a zero initial orientation (0°) for the normal operation and Scenarios 1 and 2. For Scenario 3, $\boldsymbol{p}_c(0) = [0, 0.02, \pi/6]^T$ with a nonzero initial orientation (30°) is set to highlight the reflection about the $\theta_c(0)$ line. The attack parameters in these scenarios are determined as follows.

- *Normal operation (no attack):* $\boldsymbol{S}_x = \boldsymbol{I}_3, \boldsymbol{d}_x = \boldsymbol{0}, \boldsymbol{S}_u = \boldsymbol{I}_2,$ and $\boldsymbol{d}_u = \boldsymbol{0}$.
- *Scenario 1—Reflection attack ($\beta_{11} = 1, \theta_c(0) = 0$):* $\boldsymbol{S}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$, $\boldsymbol{d}_x = \begin{bmatrix} 0 \\ 0.04 \\ 0 \end{bmatrix}$, $\boldsymbol{S}_u = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, and $\boldsymbol{d}_u = \boldsymbol{0}$.
- *Scenario 2—Scaling attack ($\beta_{11} = 0.5$):* $\boldsymbol{S}_x = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\boldsymbol{d}_x = \begin{bmatrix} 0 \\ -0.02 \\ 0 \end{bmatrix}$, $\boldsymbol{S}_u = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$, and $\boldsymbol{d}_u = \boldsymbol{0}$.
- *Scenario 3—Reflection attack with nonzero initial orientation angle ($\beta_{11} = 1, \theta_c(0) = \pi/6$):* $\boldsymbol{S}_x = \begin{bmatrix} 0.5 & \sqrt{3}/2 & 0 \\ \sqrt{3}/2 & -0.5 & 0 \\ 0 & 0 & -1 \end{bmatrix}$, $\boldsymbol{d}_x = \begin{bmatrix} -0.01\sqrt{3} \\ 0.03 \\ \pi/3 \end{bmatrix}$, $\boldsymbol{S}_u = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, and $\boldsymbol{d}_u = \boldsymbol{0}$.

Three trials of the robot's tracking of an identical desired trajectory under each FDIA scenario are reported. Fig. 7 shows the desired trajectory affected by Scenarios 1 and 2 for comparison. Fig. 8 shows different control commands received by the robot as it completes a sinusoidal path. When not under any attack, linear velocity converged to around 0.02 m/s, and angular velocity showed a sinusoidal pattern with a period of 4 s. The red and blue lines, respectively, depict the commands received by the robot under attack in Scenario 1 (reflection) and Scenario 2 (scaling).

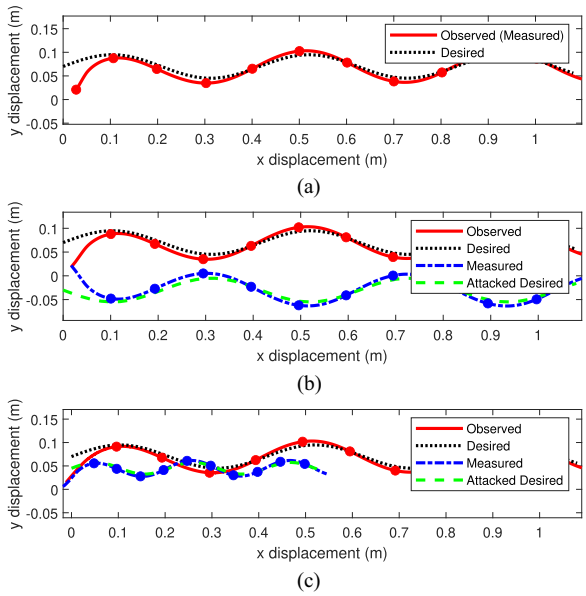



Fig. 7. Robot position compared to position perceived by controller. Controller observation is identical to the (a) original trajectory, but the actual measured position of the robot follows a modified desired path in the (b) reflection (scenario 1), and (c) scaling (scenario 2) attack scenarios. Solid circles mark the robot's position at 1-s intervals.

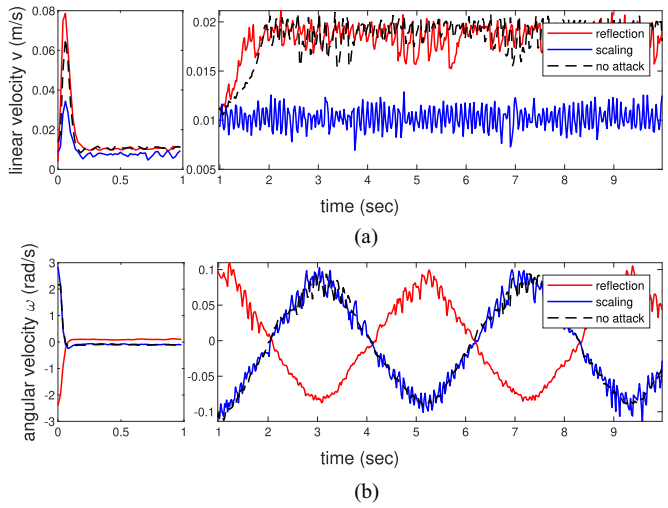



Fig. 8. Control commands received by mobile robot in Scenarios 1 and 2. (a) Linear velocity command affected in the scaled attack. (b) Angular velocity affected after the reflection attack.

In Scenario 1, the linear velocity command showed the same tendency as the base case, while the angular velocity command was reflected. In contrast, angular velocity command remained in phase with the base case in Scenario 2, but linear velocity converged to 0.01 m/s, following the scaling factor chosen beforehand. This change in input commands resulted in reflection and scaling of the actual trajectory of the robot shown in Fig. 7.

The perfectly undetectable attack is carried out in the feedback loop according to the above scenarios, leading to observed positions that do not reflect the robot's current state. Fig. 7 also shows the comparison between the actual trajectory and

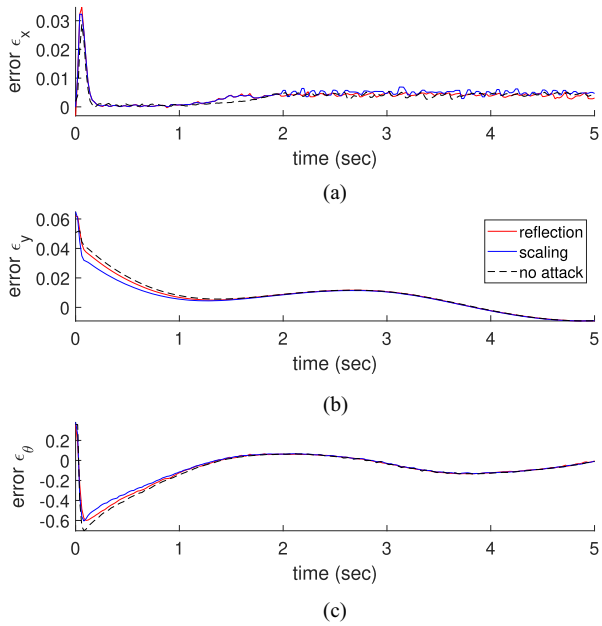



Fig. 9. Error dynamics as observed by the controller for Scenarios 1 and 2. (a)–(c) $x$, $y$, and $\theta$ error used in the controller, respectively. The overall error dynamics stayed the same regardless of the attack.

the observed position perceived by the separate controller. After the attack is applied, the observed position of the robot matches very well with the desired trajectory, as in Fig. 7. The controller observed that the robot is converging well to the predefined desired trajectory in black. This observation aligns with Fig. 9, which shows the error dynamics observed by the controller. However, the robot's actual position is at the measured position in blue, which follows the attacked desired trajectory in green. Even when system behavior is significantly changed through FDIA, there are no apparent signs of such deviation from the intended trajectory in the error dynamics to characterize the application of such an attack. Successful application of a residual-based detection method [6] is unlikely in the attack scenarios presented. Fig. 10 shows the measured position of the robot in each scenario as observed from an overhead position.

In addition, Fig. 11 shows Scenario 3 (reflection attack with a nonzero orientation angle) for a clearer visual representation of the reflection attack. The robot's initial condition $\boldsymbol{p}_c(0)[0,\ 0.02,\ \pi/6]^T$ was used. A nonzero initial orientation $\theta_c(0) = \pi/6$ sets the axis of reflection, as in Fig. 5(a).

## VI. Perfectly Undetectable FDIA Resilient State Monitoring

### A. Affine Transformation Resilient SMSFs

Based on the assumption of affine transformation-based perfectly undetectable FDIA described above, the presence of nontrivial attack matrices $\boldsymbol{S}_x$, $\boldsymbol{d}_x$, $\boldsymbol{S}_u$, and $\boldsymbol{d}_u$ that realize perfectly undetectable FDIAs has been demonstrated. *Proposition 1* is a very strong condition that makes it theoretically impossible for the controller to detect an attack based on the observation of compromised observables $\tilde{\boldsymbol{p}}_c$ corresponding to command $\boldsymbol{u}$.

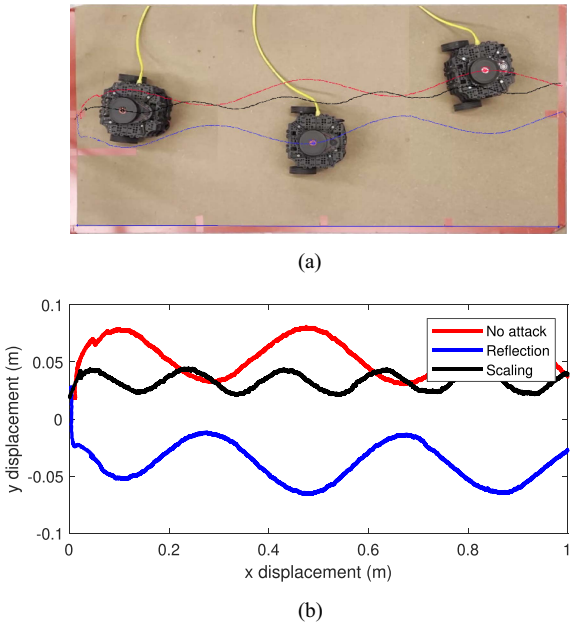



Fig. 10. Trajectory of robot acquired through video analysis (see Scenarios 1 and 2). (a) Overlaid experimental screenshots. (b) Acquired robot trajectories.

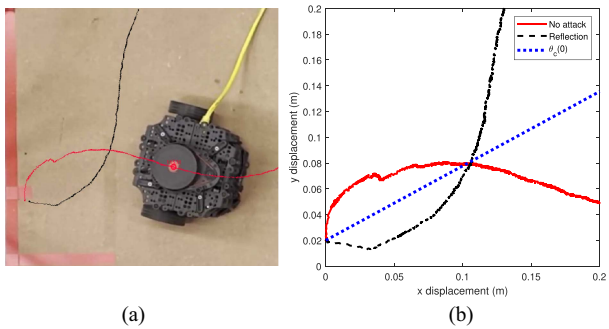



Fig. 11. Reflection about nonzero initial orientation of $\pi/6$ rad (see Scenario 3): (a) overhead video and (b) video analysis with orientation shown in blue.

The proposed countermeasure is to implement a separate function $\Phi(\boldsymbol{x})$ for state monitoring in the plant, evaluated based on the ground truth states, and compare its counterpart evaluated in the controller, as illustrated in Fig. 12. Any discrepancies between them that exceed an acceptable level of noise could indicate a possible attack. In the literature, several methods to detect FDIAs have been proposed, e.g., [6], [9], and [32]. It should be noted that, in contrast to conventional studies in the literature, this work assumes that the communication channel transmitting the output of the signature function is also susceptible to affine transformation FDIA with attack matrices $\boldsymbol{S}_\Phi$ and $\boldsymbol{d}_\Phi$. As shown in Fig. 12, the attacker might determine $\boldsymbol{S}_\Phi$ and $\boldsymbol{d}_\Phi$ based on the eavesdropping of $\boldsymbol{x}$.

The proposed SMSF serves as an authentication method similar to auxiliary systems [25] and hash functions [27]. In contrast with a hash function, the SMSF can be tailored to suit the control system under operation; for instance, an SMSF can be formulated to accommodate varying state dimensions. The output of an SMSF can be designed to be smooth, unlike that

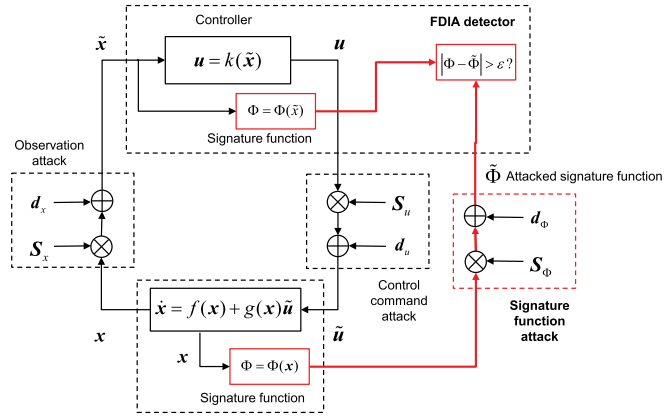



Fig. 12. Continuous state monitoring by using a signature function under affine transformation-based FDIA.

of a hash function. This smoothness allows the SMSF to be more interpretable along a smooth trajectory in the presence of noise. The SMSF is also a static function that does not require stabilization, contrary to dynamic auxiliary systems. The static design enhances resilience against certain attacks and simplifies implementation. These features collectively create a robust mechanism for detecting FDIAs.

The proposed SMSF is constructed to be resilient to both scaling and reflection FDIAs, as described in Appendix B.

*Proposition 4 (Scaling and reflection attack resilient SMSF):* As analyzed in the following in detail, the SMSF must be injective, nonlinear, and noninvertible. The noninvertibility may be achieved by choosing a dimensional reduction function, such as a scalar function, which takes multiple inputs. Suppose a scalar signature function $\Phi(\boldsymbol{x})$ of the state $\boldsymbol{x}$. If $\Phi(\boldsymbol{S}_x\boldsymbol{x}+\boldsymbol{d}_x) = \boldsymbol{S}_\Phi\Phi(\boldsymbol{x})+\boldsymbol{d}_\Phi$ holds only when $\boldsymbol{S}_\Phi = 1$ and $\boldsymbol{d}_\Phi = 0$, then the function $\Phi(\boldsymbol{x})$ is appropriate as an FDIA-resilient SMSF at least by affine transforms.

*Remark 6 (Inappropriateness of linear functions for attack-resilient state monitoring):* Note that a linear function is not appropriate at all, including the integration of inputs (total control effort), as the input-to-output relationship is linear; therefore, a scaling attack ($\beta$ and $1/\beta$ combination; see Appendix B, linear functions) is always applicable. Also, a linear signature function may be easily estimated by standard least-squares estimation techniques, necessitating that the function be nonlinear to resist FDIA.

### B. Construction of Signature Functions for Continuous State Monitoring

Consider a positive-definite function as a candidate signature function: $\Phi(\boldsymbol{x}) > 0, \boldsymbol{x} \neq 0, \Phi(\boldsymbol{x}) = 0, \boldsymbol{x} = 0, \boldsymbol{x} \subset S \in R^n$ where $S$ is an operational range of $\boldsymbol{x}$. This nonnegative property enforces the additive attack $\boldsymbol{d}_\Phi = 0$, otherwise, it is detectable. Note that $\boldsymbol{x}(0) \neq \boldsymbol{0}$ may be used as long as the positive definiteness is achieved.

*Proof:* $\tilde{\Phi}(\boldsymbol{x}) = \boldsymbol{S}_\Phi\Phi(\boldsymbol{x})+\boldsymbol{d}_\Phi = 0$ holds if and only if $\boldsymbol{d}_\Phi = 0$. ■

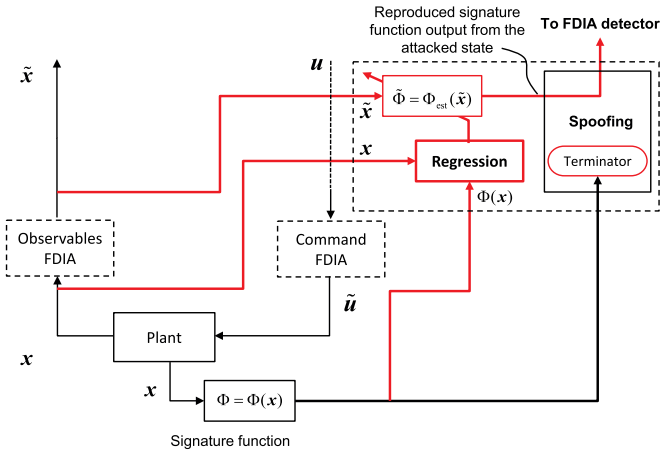



Fig. 13. Spoofing attack to state monitoring via adversarial regression of signature function.

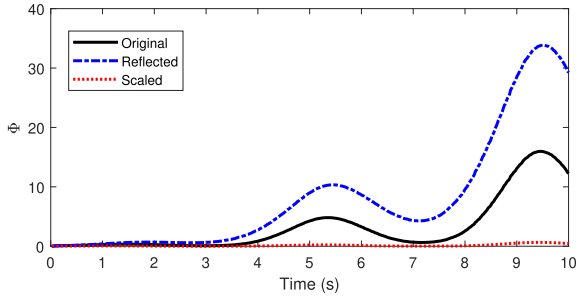



Fig. 14. State monitoring by using an example signature function $\tilde{\Phi}(x, y)$ evaluated at the plant under perfectly undetectable attacks: Scenario 1 (reflection) and Scenario 2 (scaling) compared to $\Phi(\tilde{x}, \tilde{y})$ evaluated at the controller.

Although it is a nonlinear function, a polynomial function $\Phi(x, y) = x^2 + y^2$ is not appropriate since the function is known to have the *homogeneity of degree two*, i.e., $\Phi(\alpha x, \alpha y) = \alpha^2 x^2 + \alpha^2 y^2 = \alpha^2 \Phi(x, y)$. The attacker can introduce an FDIA that multiplies the output of the signature function by the square of the scaling factor, rendering the attack undetectable. Also, this candidate function only concerns the distance from the origin, $\sqrt{x^2 + y^2}$ exhibits radial symmetry about the origin and therefore deemed inappropriate.

Without loss of generality, we can consider constructing a signature function that is 1) positive-definite (taking 0 only at the origin) to detect linear translation attacks by $\boldsymbol{d}_\Phi$, 2) non-invariant under scaling and lacking symmetry to detect scaling and reflections by $\boldsymbol{S}_\Phi$. Although guaranteeing the detection of general affine transformations is challenging, roughly speaking, 3) having asymmetric contours (or level sets) would be necessary. In the following, a polynomial function of the state variables is considered as a candidate signature function, constructed according to the following design guidelines.

1) Use even-powered terms (with at least two different powers) to ensure nonnegativity and avoid homogeneity of any specific degree. Unlike Lyapunov or "Lyapunov-like" functions used in the literature, the negative (semi)definiteness of the derivative of the function is not strictly necessary for monitoring purposes.
2) Incorporate odd-powered terms to ensure that sign flips (e.g., $x^3$ and $(-x)^3$) compute differently. Note that odd-powered terms must be introduced as a part of even-powered terms to ensure nonnegativity.
3) Include coupled terms between different variables to introduce asymmetry (e.g., $x^2 y$). Note that these asymmetries must be sufficiently nonlinear to prevent reversal through a linear transformation.

Consider a candidate signature function that involves two variables $x$ and $y$ and extends up to the quartic degree. According to Requirement 1, $x^2, x^4, y^2$, and $y^4$ must be included. According to Requirements 2 and 3, some (or all) terms, such as $x^3, y^3, xy^2$, and $x^2 y$, should be included. For those reasons, functions of only up to the quadratic degree are not suitable. Note that one of the state variables $\theta$ is not used for simplicity. While $\Phi$ is positive semidefinite in this case, because the attacker is not able to implement a pure rotation attack about the initial position along a trajectory, this property does not impact the ability of state monitoring.

*Remark 7 (Constructing more complex signature functions):* To enhance resilience against estimation attacks, one can consider incorporating a variety of mathematical constructs beyond simple polynomials. These may include exponential functions for rapid nonlinear growth, discontinuous functions for abrupt changes, and piecewise continuous functions to address multiple symmetries within heterogeneous robot teams. A well-informed choice for the design of the SMSF should not possess the same symmetry as the target plant dynamics. When constructing such functions, the defender should balance complexity with computational efficiency.

In another attack scenario, an intelligent attacker can estimate the signature function through regression to fully reproduce $\Phi(\boldsymbol{x})$ and completely alter the signal to align with $\tilde{\Phi}$, as shown in Fig. 13, making the attack remain undetectable. This implies that the SMSF remains secure only until an attacker who can intercept the state variables $\boldsymbol{x}$ and $\Phi(\boldsymbol{x})$ fully estimates the function, and the security may be quantified by its sampling complexity. The state monitoring approach will lose all effectiveness immediately if $\boldsymbol{\Phi}(\boldsymbol{x})$ is fully known by the attacker.

When implementing the SMSF approach, it is assumed that the structure of the function, such as being a polynomial function of the state variables and its degree, may be known. However, the coefficients are not known at the beginning of system operation. It is safe to change the coefficients before each system operation as a *moving target*. However, the coefficients remain fixed during each operation, as changing them would require additional communication between the controller and plant that may be intercepted.

$\Phi(x, y)$ may be estimated by using polynomial regression (PR), Gaussian process regression (GPR), and neural network regression, or other alternatives. The Vapnik-Chervonenkis (VC) dimension [33] of $\Phi(x, y)$ is a reasonable starting point for estimation. However, this should be considered a minimum guideline; more samples may be beneficial for robust estimation, particularly in complex regions of the function domain. The sample complexity [34], [35], which quantifies the number of examples needed to learn a function to a given accuracy,

increases with the VC dimension and the desired precision of the estimate. In practice, the required number of samples can be significantly higher than this lower bound, especially for complex, nonlinear functions. As demonstrated in the illustrative example in the following, the estimation of $\Phi(x, y)$ only from intercepted samples along the realized trajectory is much more challenging for the attacker, further increasing the effective sample complexity.

*Definition 2 (Security of state monitoring along a trajectory against adversarial estimation):* Let $\hat{\Phi}(\boldsymbol{x})$ be the function estimated by the attacker, using $N$ intercepted samples $\boldsymbol{x}(t_i), i = 1, \ldots, N$, collected along the trajectory from $t = 0$ to the current time, where $0 \leq t_1 < t_2 < \ldots < t_N$. The security of the SMSF is maintained if

$$\sup_{\boldsymbol{x} \in S} ||\boldsymbol{\Phi}(\boldsymbol{x}) - \hat{\boldsymbol{\Phi}}(\boldsymbol{x})|| > \epsilon > 0$$

where $S \subset \mathbb{R}^n$ is the relevant state space. If this condition holds, then the attacker cannot alter $\boldsymbol{\Phi}(\boldsymbol{x})$ being sent from the plant to the controller below a threshold $\epsilon$ throughout the state space, and therefore any attack can be detected by the controller.

*Remark 8:* As an additional security measure, it is recommended to encrypt $\Phi$ and evaluate it using methods, such as homomorphic encryption, applicable to real-time control [36], [37]. It is expected that the security of the signature function will improve in accordance with the sample complexity in both the cryptosystem and the target plant dynamic model [35]. However, note that the application of encryption does not fully prevent the risk of signature function estimation.

### C. Illustrative Example: Adversarial Estimation Scenarios

A quartic, scalar signature function can be constructed according to the guidelines in Section VI-B as

$$\begin{aligned}\Phi(x, y) &= x^4 + y^4 + (x - 50xy)^2 + (xy - 5y)^2 \\ &= x^4 + y^4 + x^2 + 25y^2 - 100x^2y - 10xy^2 \\ &\quad + 2501x^2y^2 \end{aligned} \tag{48}$$

for which attack parameters $\boldsymbol{S}_\Phi$ and $\boldsymbol{d}_\Phi$ such that $\boldsymbol{S}_\Phi \Phi(\boldsymbol{x}) + \boldsymbol{d}_\Phi = \Phi(\tilde{\boldsymbol{x}})$ do not exist, and the affine transformation attack on the signature function shown in Fig. 12 is impossible. With this signature function, under the perfectly undetectable attacks *Scenarios 1 and 2* in Section V, for example, when the communication line is not compromised, i.e., $S_\Phi = 1, d_\Phi = 0$, $\Phi(\tilde{\boldsymbol{x}})$ evaluated at the controller and $\tilde{\Phi}(\boldsymbol{x})$ evaluated at the plant along the attacked trajectories are shown in Fig. 14. In this specific case, the scaling attack affects the $\Phi$ values more drastically as the magnitude of the robot's position sees greater changes than those of the reflection case, leading to the detection of the attacks. As a result, a significant disparity is indicative of the attack on the system, and only when no attack is performed, $\Phi(\tilde{\boldsymbol{x}}) = \tilde{\Phi}(\boldsymbol{x})$ holds.

Next, as an alternative strategy for the attacker, they attempt to directly estimate $\Phi$ through collected data. The complexity of such adversarial estimation for the example signature function given in (48) is evaluated along the trajectories demonstrated in *Scenarios 1 and 2* in Section V. As representative regression techniques, PR and GPR are applied. For PR, it is plausible to assume that the attacker knows the degree of the polynomial, and thus all the polynomial bases are known. The estimation is then performed to identify the coefficients. The main challenge for the attacker lies in their ability to estimate $\Phi$ (denoted as $\hat{\Phi}$) by intercepting the attacked trajectory $\boldsymbol{x}$ with sufficient accuracy to reproduce $\Phi(\tilde{\boldsymbol{x}})$. The success of this adversarial estimation and reproduction is critical, as it makes the state monitoring process based on $\Phi$ fully ineffective. Since the VC dimension of (49) is 15, we are seeing if using 150 sample points ($15 \times 10$) for 3 s, follows the heuristic of using about ten times the VC dimension. Here, the attacker is assumed to eavesdrop for the first 3 s ($t = 0 - 3$) for each of the operations and then uses the acquired $\hat{\Phi}$ to predict the following values. In this instance, the attacker allows the controller to monitor $\Phi(\tilde{\boldsymbol{x}})$ till $t = 3$ using samples limited to the trajectory, resulting in poorer estimations. Fig. 15 shows the comparison between $\hat{\Phi}(\tilde{\boldsymbol{x}})$ and $\Phi(\tilde{\boldsymbol{x}})$ over time. As the sample is not distributed evenly within the workspace, only along the realized trajectories, the estimation fails, and the security of the monitoring approach is maintained.

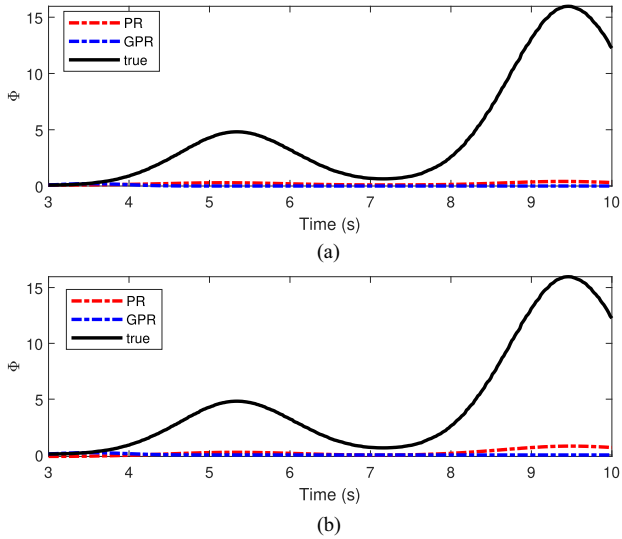



Fig. 15. Unsuccessful adversarial estimation $\hat{\Phi}(\tilde{\boldsymbol{x}})$ with 150 samples for (a) Scenario 1 (reflection) and (b) Scenario 2 (scaling).

From the attacker's perspective, there may be two potential attempts to improve the performance of $\hat{\Phi}$: 1) an increased number of samples and 2) a trajectory that provides a wider coverage of the workspace. This allows for a bigger window for the controller to detect the presence of an FDIA before $\hat{\Phi}$ can be estimated. Fig. 16 shows the performance of the estimation with increased numbers of samples for regression. For the first 500 samples, the experimental data from Section V were used, followed by the data generated by simulation. Normalized root-mean-square error (NRMSE) is shown as the metric of fitness. In general, increasing the number of samples improved the performance of estimation. Nevertheless, neither of the attack scenarios provides perfect estimation, probably due to insufficient coverage of the workspace along the attacked trajectories. If the user operates the robot along a trajectory that widely explores the workspace, then the attacker would collect a sufficiently rich dataset. Fig. 17

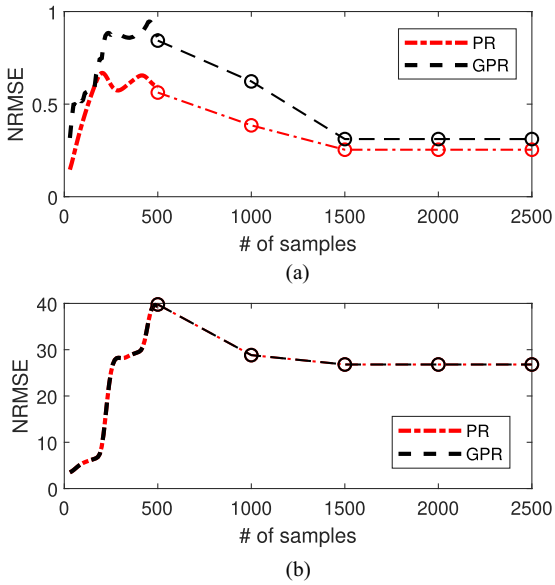



Fig. 16. Performance of signature function estimation: Normalized RMSE with respect to the number of samples $N$ with experimental and simulated data: (a) for Scenario 1 (reflection) and (b) Scenario 2 (scaling). Simulated data displayed for points beyond 500 samples.

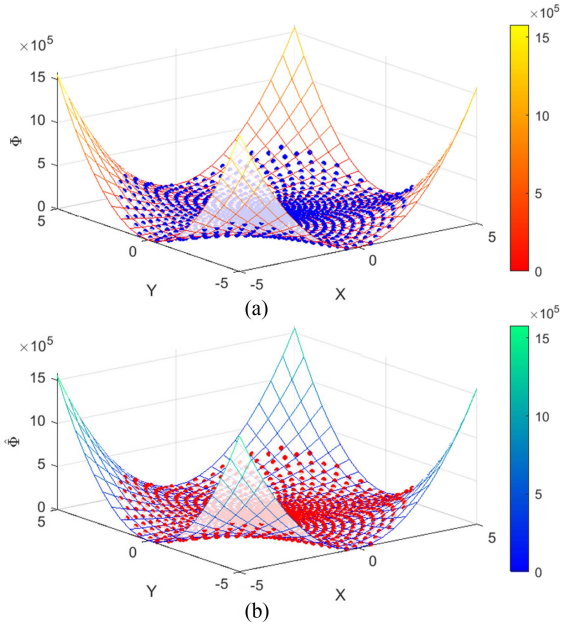



Fig. 17. PR over a large number of samples $N=1000$ with a simulated spiral trajectory with a wide coverage of the workspace: (a) true $\Phi$ and (b) estimated $\hat{\Phi}$ with an NRMSE value of 0.059.

shows the performance of PR performed with data from a fictitious spiraling trajectory simulated up to 1000 samples. With a sample size of 150, the recommended sample size based on the VC dimension, the estimation result was significantly better compared to the previously considered sinusoidal trajectory with an NRMSE of only 0.26. The distribution of samples was observed to be the main factor for successful estimation.

## VII. Discussion

Identification and classification of potential cyberattacks are important tasks when working with networked robots, as a thorough understanding of the effects and conditions then allows for further studies on defense strategies. Preexisting discussions of deceptive and undetectable attacks are often limited, with strong assumptions restraining the attacker's capability and focusing on simple systems [10], [25]. However, even without conditions, such as full knowledge of system dynamics or linearity, substantial attacks could be applied to systems.

The mobile robot tracking control experiment in this article demonstrates that for an attacker capable of compromising both legs of a networked control system, the only information required by the attacker was the structure of the Jacobian $\boldsymbol{J}$ and initial conditions, unlike covert attacks that necessitate perfect knowledge of the plant dynamics. In the studies by Sandberg et al. [6] and Zhai et al. [38], the realization of covert attacks is formulated as an optimization problem using the plant dynamic model. These studies allow for control-theoretic discussion and are thus of academic interest, but their implementation may be challenging from an engineering perspective .

In contrast, affine transformation seems to be a more practical attack method. Relatively simplistic attacks with constant attack parameters, which do not alter the degree of the closed-loop dynamics, were effective to significantly modify the robot's behavior while remaining undetectable from the controller's perspective. The affine transformation-based FDIAs are highly effective on multidimensional robotic systems rather than low-dimensional systems but with a high degree. By manipulating variables of the same physical quantities through operations, such as scaling or reflection, attackers can significantly alter a robot's behavior while maintaining mathematical consistency. This attack exploits the multidimensional nature of robotic systems, making attacks difficult or impossible to detect yet simple to implement, even with partial system knowledge. The preservation of mathematical structures in these multidimensional spaces poses significant challenges for traditional detection methods.

Even when a high degree or nonlinear dynamics are concerned, it is common to delegate the low-level control to the robot and employ simpler feedback loops over the network. For example, the Turtlebot system used in this article delegated low-level motor control to the robot while high-level control was performed by the remote server. Such a hierarchical control architecture may increase the risk of affine transformation-based attacks or other types of FDIAs. It should also be noted that modern mobile robot systems increasingly use distributed architectures where robots communicate primarily with neighbors and share only limited task-related information rather than full state data. In such cases the applicability of our attack methodology remains uncertain and requires future investigation.

Regarding electronic watermarking [39], even if white noise is added at the controller, the observed dynamics remain unchanged, as shown in (26), so the effect of white noise will be accurately restored on the observation side. This means that the covariance of the estimation error by the observer does

not change, making existing watermarking and probing signal methods ineffective.

Note that the existence of perfectly undetectable FDIAs is guaranteed for linear plants, and their extent varies based on system knowledge [11], [21] but not necessarily for all nonlinear dynamic systems. While *Proposition 1* provides a general framework for nonlinear dynamics in affine form, the existence of solutions depends on the specific system characteristics. This article demonstrates that such attacks exist for mobile robot dynamics, a significant finding in robotic security.

However, fully generalizing perfectly undetectable FDIA to all nonlinear systems requires further research, as conditions for their existence may vary widely across different nonlinear plants. A relevant concept is known as equivariant dynamics [40]. Characterization of attacks as transformation groups that preserve properties of vector fields would allow a more general classification of each attack and consequently its detection. The authors are exploring the use of Lie groups for this purpose [22].

As a countermeasure against perfectly undetectable FDIAs, a state monitoring approach using a signature function has been proposed. A sufficiently complex polynomial function is resilient against affine transformations. However, it should be noted that this approach has partial vulnerability to estimation attacks; given enough time and data, an attacker could potentially estimate the signature function through regression techniques, especially if they can observe a trajectory that covers a wide range of the workspace. The SMSF approach has been experimentally validated as an effective deterrence against undetectable FDIA when used with additional detection methods in the authors' other work [41].

In addition, the example signature function discussed and analyzed in this article is merely satisfactory. Although the example SMSF was shown to be practically safe even while assuming the worst-case scenario by placing no restrictions on the attacker's resources, it is understandable that estimation of the SMSF is eventually achieved by the attacker, indicating the necessity of frequent updates of the function before this happens. Future works should discuss dynamic management of SMSF wherein the attacker's resources and difficulty of SMSF's estimation should be balanced with system performance in a game-type optimization [3], [42]. The optimal formulation of the signature function and its update frequency will be explored in our future work.

## VIII. Conclusion

The article focuses on a mobile robot trajectory tracking control system as a case study, highlighting the susceptibility of nonlinear systems with partially linear dynamic properties and symmetries to this type of attack. The experimental results using a Turtlebot 3 platform validate the practicality of implementing such attacks, emphasizing the urgent need for more robust security measures in cyber-physical systems (CPS). This article demonstrated that a typical mobile robot trajectory tracking control system is susceptible to perfectly undetectable FDIAs. Two specific types of perfectly undetectable FDIA are possible: scaling and reflection attacks, both based on affine transformations. These findings demonstrate the critical need for more robust detection mechanisms and resilient control strategies to protect such systems.

Future work will focus on developing effective countermeasures to mitigate the risks associated with these sophisticated cyberattacks and enhance system security in real-world applications, including customization of SMSFs. In addition, exploring response strategies that leverage machine learning could offer promising avenues for advancing the resilience of CPS against increasingly complex attack vectors. Furthermore, the introduction of time-variant perfectly undetectable attacks could lead to more sophisticated and powerful attacks compared to the simple scenarios mentioned in this article.

## Appendix A
## Perfectly Undetectable FDIA From the Plant's Perspective

*Definition A1: Perfectly undetectable FDIA from the plant's perspective* (see [6], [23], and [24]). Let $y(x(0), u, a)$ denote the response of the system for the initial condition $x(0)$, input $u(t)$, and attack signal $a(t)$. The attack is perfectly undetectable if

$$y(x(0), u, a) = y(x(0), u, 0), t \geq 0. \tag{49}$$

The attacker leaves no traces in the measurements of $y$. Consequently, the attacker can impact the system's performance or behavior without being detected by an attack detector that utilizes $y$ for attack detection. Research has shown that (49) can be achieved through zero dynamics attacks in the presence of transmission zeros [32].

## Appendix B
## Linear FDIA Vulnerability in Polynomial and Trigonometric Functions

Consider a scalar function $g(x)$ with scalar attack parameters $\alpha(\neq 0)$ affecting the output and $\beta(\neq 0)$ affecting the input, resulting in the function $\tilde{g}(x) = \alpha g(\beta x)$. Examine conditions where $g(x) = \tilde{g}(x)$ holds for all $x$. The function is said to be susceptible to linear attacks if nontrivial solutions for $\alpha$ and $\beta$ exist other than the trivial case (i.e., $\alpha = \beta = 1$). Results for representative functions and brief proofs are presented in the following.

*Proposition B1: Linear FDIA vulnerability of representative scalar functions.*

1) $g(x) = cx$: $\alpha g(\beta cx) = \alpha\beta cx$: There are an infinite number of solutions that satisfy $\alpha\beta = 1$ for such linear functions. Note that the attacker does not require knowledge of the coefficient $c$.
2) $g(\theta) = \cos(\theta)$: Consider its first and second derivatives with respect to $\theta$: $g'(\theta) = -\sin(\theta) = -\alpha\beta\sin(\beta\theta)$, and $g''(\theta) = -\cos(\theta) = -\alpha\beta^2\cos(\beta\theta)$. $\beta = \pm 1$ since $\beta^2 = 1$. When $\beta = -1$ (nontrivial case), $\alpha = -1$ since $\alpha\beta = 1$.
3) $g(\theta) = \sin(\theta)$: Similar analysis to the above yields $\beta = -1$ (nontrivial case) and $\alpha = 1$ since $\alpha\beta = -1$.
4) $g(x) = cx^2$: $\alpha g(\beta x) = \alpha c\beta^2 x^2$: There are an infinite number of solutions that satisfy $\alpha\beta^2 = 1$. Note that the attacker does not require knowledge of the coefficient $c$.

5) $g(x) = ce^x$: $\alpha g(\beta x) = \alpha c e^{\beta x}$: Comparing the first derivative functions with respect to $x$: $g'(x) = ce^x = \alpha c \beta e^{\beta x}$ yields only the trivial case, $\alpha = \beta = 1$. The exponential function is resistant to linear attacks.

## REFERENCES


[1] D. G. Pivoto, L. F. de Almeida, R. da Rosa Righi, J. J. Rodrigues, A. B. Lugli, and A. M. Alberti, "Cyber-physical systems architectures for industrial Internet of Things applications in industry 4.0: A literature review," *J. Manuf. Syst.*, vol. 58, pp. 176–192, 2021.

[2] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419839596. [Online]. Available: https://doi.org/10.1177/1729881419839596

[3] Y. Jiang, S. Wu, R. Ma, M. Liu, H. Luo, and O. Kaynak, "Monitoring and defense of industrial cyber-physical systems under typical attacks: From a systems and control perspective," *IEEE Trans. Ind. Cyber- Phys. Syst.*, vol. 1, pp. 192–207, 2023.

[4] Y. Li, D. Shi, and T. Chen, "False data injection attacks on networked control systems: A Stackelberg game analysis," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3503–3509, Oct. 2018.

[5] Y. Dong, N. Gupta, and N. Chopra, "False data injection attacks in bilateral teleoperation systems," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 1168–1176, Mar. 2020.

[6] H. Sandberg, V. Gupta, and K. H. Johansson, "Secure networked control systems," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 5, no. 1, pp. 445–464, 2022. [Online]. Available: https://doi.org/10.1146/annurev-control-072921-075953

[7] Y. Mao, H. Jafarnejadsani, P. Zhao, E. Akyol, and N. Hovakimyan, "Novel stealthy attack and defense strategies for networked control systems," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3847–3862, Sep. 2020.

[8] S. E. McLaughlin et al., "The cybersecurity landscape in industrial control systems," *Proc. IEEE*, vol. 104, pp. 1039–1057, 2016.

[9] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.

[10] R. S. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 90–95, 2011.

[11] J. Ueda and J. Blevins, "Affine transformation-based perfectly undetectable false data injection attacks on remote manipulator kinematic control with attack detector," *IEEE Robot. Autom. Lett.*, vol. 9, no. 10, pp. 8690–8697, Oct. 2024.

[12] S. Gao, H. Zhang, Z. Wang, and C. Huang, "A class of optimal switching mixed data injection attack in cyber-physical systems," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1598–1605, Feb. 2021.

[13] C. Wu et al., "A secure robot learning framework for cyber attack scheduling and countermeasure," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3722–3738, May 2023.

[14] S. Amin, A. A. Cárdenas, and S. S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *Proc. 12th Int. Conf. Hybrid Syst.: Computation Control*, 2009, pp. 31–45.

[15] A. Petrillo, A. Pescape, and S. Santini, "A secure adaptive control for cooperative driving of autonomous connected vehicles in the presence of heterogeneous communication delays and cyberattacks," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1134–1149, Mar. 2020.

[16] W.-H. Chen, J. Yang, L. Guo, and S. Li, "Disturbance-observer-based control and related methods—An overview," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1083–1095, Feb. 2015.

[17] M. Schulze Darup, A. B. Alexandru, D. E. Quevedo, and G. J. Pappas, "Encrypted control for networked systems: An illustrative introduction and current challenges," *IEEE Control Syst. Mag.*, vol. 41, no. 3, pp. 58–78, Mar. 2021.

[18] H. B. Kwon, S. Kosieradzki, J. Blevins, and J. Ueda, "Encrypted coordinate transformation via parallelized somewhat homomorphic encryption for robotic teleoperation," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron. (AIM)*, 2023, pp. 228–233.

[19] K. Teranishi and K. Kogiso, "Control-theoretic approach to malleability cancellation by attacked signal normalization," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 297–302, 2019.

[20] D. Boneh, G. Segev, and B. Waters, "Targeted malleability: Homomorphic encryption for restricted computations," in *Proc. 3rd Innovations Theor. Comput. Sci. Conf.*, 2012, pp. 350–366.

[21] J. Ueda, "Affine transformation-based perfectly undetectable false data injection attacks from controller's perspective on state- and output feedback linear control systems," *IEEE Trans. Ind. Cyber- Phys. Syst.*, vol. 3, pp. 656–664, 2025.

[22] H. Kwon et al., "Perfectly undetectable false data injection attacks on encrypted bilateral teleoperation system based on dynamic symmetry and malleability," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2025, pp. 16457–16463.

[23] J. Milošević, A. Teixeira, K. H. Johansson, and H. Sandberg, "Actuator security indices based on perfect undetectability: Computation, robustness, and sensor placement," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3816–3831, Sep. 2020.

[24] S. Gracy, J. Milošević, and H. Sandberg, "Security index based on perfectly undetectable attacks: Graph-theoretic conditions," *Automatica*, vol. 134, 2021, Art. no. 109925.

[25] C. Schellenberger and P. Zhang, "Detection of covert attacks on cyber-physical systems by extending the system dynamics with an auxiliary system," in *Proc. IEEE 56th Annu. Conf. Decis. Control* (CDC), 2017, pp. 1374–1379.

[26] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Robot. Auto.*, 1990, pp. 384–389.

[27] R. C. Merkle, A Certified Digital Signature, ser, *Lecture Notes in Computer Science*, vol. 435. Berlin, Germany: Springer, 1989, pp. 218–238.

[28] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Amesterdam, The Netherlands: Elsevier, 2013.

[29] P. Hartman, *Ordinary differential equations*. Philadelphia, PA, USA: SIAM, 2002.

[30] H. Zhu et al., "Secure control against multiplicative and additive false data injection attacks," *IEEE Trans. Ind. Cyber- Phys. Syst.*, vol. 1, pp. 92–100, Jan. 2023.

[31] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.* (ICRA), 2016, pp. 1271–1278.

[32] A. Hoehn and P. Zhang, "Detection of covert attacks and zero dynamics attacks in cyber-physical systems," in *Proc. Amer. Control Conf. (ACC)*, 2016, pp. 302–307.

[33] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension," *J. ACM (JACM)*, vol. 36, no. 4, pp. 929–965, 1989.

[34] R. Meyer, C. Musco, C. Musco, D. P. Woodruff, and S. Zhou, "Near-linear sample complexity for $l_p$ polynomial regression," in *Proc. 34th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA 2023)*, 2023, pp. 3959–4025.

[35] K. Teranishi, T. Sadamoto, A. Chakrabortty, and K. Kogiso, "Designing optimal key lengths and control laws for encrypted control systems based on sample identifying complexity and deciphering time," *IEEE Trans. Autom. Control*, vol. 68, no. 4, pp. 2183–2198, Apr. 2023.

[36] K. Teranishi, K. Kogiso, and J. Ueda, "Encrypted feedback linearization and motion control for manipulator with somewhat homomorphic encryption," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron. (AIM)*, 2020, pp. 613–618.

[37] J. Blevins and J. Ueda, "Encrypted model reference adaptive control with false data injection attack resilience via somewhat homomorphic encryption-based overflow trap," *IEEE Trans. Ind. Cyber- Phys. Syst.*, vol. 3, pp. 262–272, 2025.

[38] L. Zhai, K. G. Vamvoudakis, and J. Hugues, "A graph-theoretic security index based on undetectability for cyber-physical systems," in *Proc. Amer. Control Conf. (ACC)*, 2022, pp. 1479–1484.

[39] D. Du, C. Zhang, X. Li, M. Fei, T. Yang, and H. Zhou, "Secure control of networked control systems using dynamic watermarking," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13609–13622, Dec. 2021.

[40] F. Michael, "Equivariant dynamical systems," *Trans. Amer. Math. Soc.*, vol. 259, no. 1, pp. 185–205, 1980.

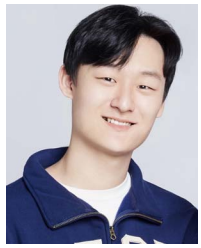
[41] H. Kwon, J. Blevins,, and J. Ueda, "Defense mechanisms against undetectable cyberattacks on encrypted telerobotic control systems," *IEEE/ASME Trans. Mechatron*, vol. 30, no. 4, pp. 2964–2971, 2025.

[42] K. Kogiso, "Attack detection and prevention for encrypted control systems by application of switching-key management," in *Proc. IEEE Conf. Decis. Control (CDC)*, 2018, pp. 5032–5037.

**Jun Ueda** (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Kyoto University, Kyoto, Japan, in 1994, 1996, and 2002, respectively.

From 1996 to 2000, he was a Research Engineer with the Advanced Technology Research and Development Center, Mitsubishi Electric Corporation, Amagasaki, Japan. From 2002 to 2008, he was an Assistant Professor with Nara Institute of Science and Technology, Ikoma, Japan. During 2005–2008, he was a Visiting Scholar and Lecturer with the Department of Mechanical Engineering, Massachusetts Institute of Technology. In 2008, he joined as an Assistant Professor with the G. W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA,, where he is currently a Professor. During 2015–2017, he was the Director for the Robotics PhD Program with Georgia Tech. He is the author of *Cellular Actuators: Modularity and Variability in Muscle-Inspired Actuation* (Butterworth-Heinemann, 2017).

Dr. Ueda is currently the Senior Editor for IEEE/ASME TRANSACTIONS ON MECHATRONICS. He was the recipient of the Fanuc FA Robot Foundation Best Paper Award in 2005, IEEE Robotics and Automation Society Early Academic Career Award in 2009, Advanced Robotics Best Paper Award in 2015, and Nagamori Award in 2021. He is a Fellow of ASME.



**Hyukbin Kwon** (Student Member, IEEE) received the B.S. degree in mechanical engineering in 2023 from the G. W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA, where he is currently working toward the Ph.D. degree in mechanical engineering.

His research interests include secure control systems and sensor fusion for robotic manipulation.