False Data Injection Attacks in Bilateral Teleoperation Systems

Yimeng Dong, *Member, IEEE*, Nirupam Gupta, *Member, IEEE*, and Nikhil Chopra, *Member, IEEE*

***Abstract*— In this brief, false data injection attacks in bilateral teleoperation systems (BTOSs) are studied, where the attacker can inject false data into the states being exchanged between the master and slave robots. To demonstrate the vulnerability of BTOS, a destabilizing false data injection attack (DFDIA) is designed and experimentally implemented. To detect any general false data injection attack, including DFDIA as a special case, a physics-based detection scheme with an encoding–decoding structure is proposed. The efficacy of the proposed attack detection scheme is demonstrated in experiments.**

***Index Terms*— Attack detection, bilateral teleoperation system (BTOS), false data injection attack.**

I. Introduction

BILATERAL teleoperation systems (BTOSs) extend the human capability to manipulate objects with the help of robotic manipulators and communication networks (Fig. 1). Such systems can be used for various tasks, including handling hazardous materials [1], space and underwater exploration [2], [3], and telesurgery [4]. One example of BTOS is shown in Fig. 1. On being manipulated by a human operator, the state (joint position q_m and velocity $\dot{q}_m$) of the master robot is transmitted to the slave robot through a communication network (wired or wireless). The slave robot is coupled to the master robot state through an appropriate controller which then guides the slave robot to complete a desired task in the remote environment. Simultaneously, the state of the slave robot is communicated to the master robot, and through the master controller, a bilateral coupling is established between the two robots.

With recent advancements in communication technologies, the Internet has emerged as a popular choice for coupling robots in teleoperation systems. However, data transfer through the Internet/wireless medium is unreliable as frequent data losses and delays are encountered, deteriorating the BTOS performance. For the past few decades, significant research efforts have been conducted to overcome these issues and they have yielded fairly successful results such as [5]–[10].

Manuscript received July 18, 2018; revised December 10, 2018; accepted February 24, 2019. Manuscript received in final form March 2, 2019. This work was supported in part by the National Science Foundation under Grant ECCS1232127 and GrantECCS1711554 and in part by the Naval Air Warfare Center Aircraft Division, Pax River, MD, USA, under Contract N00421132M022. Recommended by Associate Editor T. Hatanaka. *(Corresponding author: Yimeng Dong.)*

The authors are with the Department of Mechanical Engineering, University of Maryland, College Park, MD 20740 USA (e-mail: ymdong@umd.edu; nirupam@umd.edu; nchopra@umd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2019.2903446

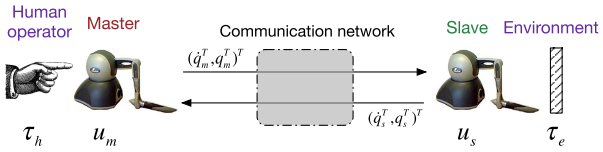

Fig. 1. Example of BTOS structure.

Besides the network effects, cyberattacks can also lead to severe disruption and damage to teleoperation systems.

In recent years, the importance of guaranteeing the safety and security of cyber-physical system (CPS) (like BTOS) is being increasingly recognized. Since most of the CPSs are safety critical, their failure can damage the critical physical system. Examples of cyberattacks include Stuxnet malware sabotaging Iran's nuclear infrastructure [11], the water SCADA system attack [12], and the power transmission network attack [13]. The traditional cyber security methods for information systems like cryptography are not adequate for cyber-physical security. The necessity of cyber-physical security study can be explained as follows. First, the CPS has real-time requirements and may additionally have energy consumption/processing power constraints. Several modern cryptographic algorithms cannot be implemented in constrained devices and cannot satisfy the real-time requirement as the algorithms are designed for desktop/server environments [14]. Second, unlike information systems, the CPS (like BTOS) involves interaction with physical systems. As a cyberattack can directly impact and even damage the critical physical system and endanger the wellbeing of operators, adequate safety nets are required. Third, the underlying physics of CPS can potentially be exploited to better understand the attack impact and to provide a different perspective on the system security from existing cryptographic tools.

The study of cyber-physical security in networked control systems has been conducted from two perspectives: 1) attack model and design and 2) defense mechanisms. Various works on these two perspectives can be found in [15] and the references therein. Particularly, for a linear descriptor system, Pasqualetti *et al.* [16] define the attack detectability and identifiability and construct an observer-based attack detection and identification monitor. Unlike the observer-based method, Eyisi and Koutsoukos [17] propose an energy-based attack detection scheme for a passive networked CPS based on the fact that the attack influences the energy balance of the system.

The synchronization problem in BTOS can be viewed as a special case of multiagent system (MAS) synchronization. Hence, the security of MAS is also relevant to the problem considered herein. As MAS can be modeled as a graph with agents as nodes and communication links as edges, the attack

on MAS can be divided into two categories. The first, termed a node attack, that can convert a normal node into a malicious node, and the other is called an edge attack, which can break link connections or modify information in links. Some works in the literature focus on node attacks and resilient MAS design [18]–[20]. Other works study the attack detection and identification methods. An observer-based malicious node detection and identification scheme for the first-order MAS consensus is proposed in [21]. An unknown input observer (UIO)-based distributed faulty node and edge detection and isolation scheme for the second-order MAS consensus are studied in [22].

The security of BTOS has attracted recent attention from the research community since the attacker has the potential to cause harm to the robots, humans, and the environment involved, which is highly undesired in any bilateral teleoperation application. The security threats in surgical telerobotics are identified in [23], and an experimental analysis of security threats on the RAVEN surgical robot is presented in [24]. However, no rigorous theoretical analysis of the attack strategies and the prevention/mitigation solutions has been discussed. Also, it is worth noting that Bonaci *et al.* [24] study a so-called "Surgeon's Intent Modification" attack, which modifies the data packet from surgeon to surgical robot. Other related works include secure communication protocol designs for telesurgery [25], [26]; however, a theoretical analysis of the attacks is not considered in this brief.

In this brief, we address a particular kind of edge attack called a false data injection attack, which can inject false data into the data content transmitted in the communication link. The false injection data attack was initially studied in [27] for a discrete linear time-invariant Gaussian system equipped with a Kalman filter and LQG controller. In the literature of BTOS security, a similar attack called "message modification and spoofing attack" was discussed in [23]. This kind of attack can first block the communication between the master and slave robots, establish a communication independently with both sides, and then forward the modified malicious messages. In our previous work [28], the design and detection of a similar attack termed "content modification attack" were studied for an MAS with linear double-integrator dynamics.

The main motivation for the security analysis in this brief is to design and detect the cyberattacks for networked robotic systems (like BTOS) by constructively utilizing the physics of the robotic system. To better understand the severe impact of the false data injection attack, a specific attack called the destabilizing false data injection attack (DFDIA) is designed based on the notion of physical storage function from the attacker's perspective. From the defender's perspective, a physics-based attack detection scheme with an encoding–decoding structure is proposed to detect a general false data injection attack. The attack mapping is made precise in Section III-B. In the proposed scheme, the original state data are first encoded using an encoding factor on the sending end, and then, the original state data are recovered by decoding on the receiving end. The proposed scheme is based on the inherent physical relation within the transmitted data: once the encoded data content is corrupted, the inherent physical relation of the decoded data is violated, thereby the attack can be detected. The proposed scheme also requires certain scalar initial values in the system to be shared securely between both sides before the operation, which is made clear in Section IV. It is anticipated that the physics-based attack detection can provide an additional layer of security for CPS (like BTOS), thereby complimenting existing cryptographic methods such as those in [25] and [26].

Compared with the detection schemes in the aforementioned works, the proposed algorithm is advantageous in certain respects. First, [21], [22], [16], and [17] consider linear system dynamics, whereas a nonlinear robotic dynamics is considered here. Thus, their detection schemes for linear systems cannot be directly applied here; second, the observer-based detection schemes [16], [21], [22] and the energy-based detection scheme [17] require the system model and are prone to high computation and communication costs, whereas the proposed attack detection scheme does not require the detailed knowledge of system parameters and is solely based on the inherent physical relation within the transmitted data. Thus, the proposed detection scheme has the following advantages: it is fast, computationally light, and does not require knowledge of system parameters. Compared with our previous work [28], here the false data injection attacks are studied for a BTOS with nonlinear robotic dynamics, and the detection scheme is modified and improved (Remark 2).

Paper Contribution: The main contributions of this brief are outlined as follows.

1) To demonstrate the severe attack impact, a systematic theoretical framework for designing DFDIA attacks in nonlinear BTOS is developed. Although the framework is formally demonstrated for the case when the master/slave robots are coupled using a simple proportional-derivative (PD)-like controller, attacks for other bilateral teleoperation control architectures can be similarly designed.
2) To safeguard the system, a physics-based attack detection scheme with an encoding–decoding structure is proposed for general false data injection attacks. The proposed algorithm can detect any general false data injection attacks, as detailed in Section IV, and which include DFDIA as special case.

This brief is an extension of a previous conference paper [29]. Compared with the conference paper, the additional contributions of this brief can be highlighted as follows.

1) A physics-based attack detection algorithm, based on the simple checking scheme in [29], is proposed for general false data injection attacks.
2) Reference [29, Th. 1] is demonstrated to be both necessary and sufficient.
3) A BTOS experiment platform consisting of two PHANToM Omni robots is developed, and the theoretical results are also studied through experiments.
4) The assumption that the BTOS is in free motion, which is utilized in [29], is relaxed, which then leads to more general results.

This brief is organized as follows. In Section II, mathematical notations used throughout this brief are described.

In Section III, the attack design is discussed for a PD-like BTOS control scheme. In Section IV, the attack detection scheme for general false data injection attacks is presented. Finally, the experimental results are presented in Section V.

II. Notations

Throughout this brief, the symbols $\mathbb{Z}$, $\mathbb{R}^n$, $\mathbb{R}^{n\times n}$, $\mathbb{R}_0^+$, and $\mathbb{R}^+$ denote the sets of positive integers, n-dimensional real-valued vectors, n-by-n matrices with real-valued elements, and sets of nonnegative and positive real numbers, respectively. $\|\cdot\|$ denotes the Euclidean norm of a vector. I_n, 0_n, and 0 denote n-by-n identity matrix, n-by-n zero matrix, and n-by-1 zero column vector, respectively. $diag(\cdot)$ denotes the diagonal matrix. To simplify the notation, unless otherwise necessary, the argument of a time-dependent signal is omitted [e.g., $\dot{q} = \dot{q}(t)$].

III. Attack Design for BTOS

In this section, to demonstrate the potential of a severe attack, a special false data injection attack called DFDIA is designed for a simple PD-like BTOS control scheme. The DFDIA can result in an unbounded growth of a physical storage function, thereby causing system instability and even physical system damage. In Section III-A, a PD-like BTOS control scheme is first considered as a simple case for the attack design. Then, in Section III-B, the DFDIA design is described in the context of BTOS.

A. PD-Like Control Scheme for BTOS

Consider a BTOS consisting of a pair of nonlinear revolute robotic manipulators coupled via a communication network, as shown in Fig. 1. Ignoring external disturbances and friction, the dynamics of n-link master and slave robots are given as [30]

$$\begin{aligned} M_m(q_m)\ddot{q}_m + C_m(q_m,\dot{q}_m)\dot{q}_m + G_m(q_m) &= \tau_h + \tau_m \\ M_s(q_s)\ddot{q}_s + C_s(q_s,\dot{q}_s)\dot{q}_s + G_s(q_s) &= \tau_s - \tau_e \end{aligned} \quad (1)$$

where subscript $i = m, s$ denotes the master and slave robots, respectively. Henceforth, subscript i will represent both master and slave robots. Here, $\ddot{q}_i, \dot{q}_i, q_i \in \mathbb{R}^n$ are the angular acceleration, velocity, and position, respectively, $M_i(q_i) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C_i(q_i,\dot{q}_i) \in \mathbb{R}^{n\times n}$ is the centrifugal and Coriolis matrix, $G_i(q_i) \in \mathbb{R}^n$ is the gravitational torque, $\tau_i \in \mathbb{R}^n$ is the robot control input, and $\tau_h, \tau_e \in \mathbb{R}^n$ are torques exerted by the human operator and the environment, respectively.

Robotic manipulators with revolute joints have the following fundamental properties due to the Lagrangian dynamics structure [30].

(P1) The inertia matrix $M_i(q_i)$ is symmetric positive definite, which is lower and upper bounded by

$$\lambda_m I_n \le M_i(q_i) \le \lambda_M I_n \quad (2)$$

where λ_m and λ_M are the positive minimum and maximum eigenvalues of $M_i(q_i)$ for all configurations q_i.

(P2) Under an appropriate definition of $C_i(q_i,\dot{q}_i)$, the matrix $\dot{M}_i(q_i) - 2C_i(q_i,\dot{q}_i)$ is skew symmetric.

In this section, to simplify the theoretical analysis of the attack design, the following assumptions are made for the BTOS.

(A1) The network condition is perfect, which means the network effects (such as time delays and data losses) are not considered in the attack design.

(A2) The human operator and environment can be modeled as passive systems

$$-\int_0^t \dot{q}_m(s)^T\tau_h(s)ds \ge -\beta_h, \quad \int_0^t \dot{q}_s(s)^T\tau_e(s)ds \ge -\beta_e \quad (3)$$

for $t > 0$ and some $\beta_h, \beta_e \in \mathbb{R}_0^+$.

(A3) The gravitational torques are precompensated such that

$$\tau_m = u_m + G_m(q_m), \quad \tau_s = u_s + G_s(q_s) \quad (4)$$

in the given dynamics (1). This reduces the overall dynamics of the BTOS as

$$\begin{aligned} M_m(q_m)\ddot{q}_m + C_m(q_m,\dot{q}_m)\dot{q}_m &= \tau_h + u_m \\ M_s(q_s)\ddot{q}_s + C_s(q_s,\dot{q}_s)\dot{q}_s &= u_s - \tau_e. \end{aligned} \quad (5)$$

As shown in Fig. 1, the data consisting of velocity and position signals $(\dot{q}_i^T, q_i^T)^T$ are transmitted between the two robots, and the received data are denoted by $(\tilde{\dot{q}}_i^T, \tilde{q}_i^T)^T$.

When there is no attack ($\tilde{\dot{q}}_i = \dot{q}_i, \tilde{q}_i = q_i$), similar to the analysis in [31, Th. 3.5] and [8, Proposition 2], we can adopt the following PD-like controller for BTOS:

$$\begin{aligned} u_m &= -K_d(\dot{q}_m - \tilde{\dot{q}}_s) - K_p(q_m - \tilde{q}_s) - K_{dm}\dot{q}_m \\ u_s &= -K_d(\dot{q}_s - \tilde{\dot{q}}_m) - K_p(q_s - \tilde{q}_m) - K_{dm}\dot{q}_s \end{aligned} \quad (6)$$

where $K_d, K_p, K_{dm} \in \mathbb{R}^+$, thereby the following statements hold.

1) The position error $e_q := q_s - q_m$ is bounded, and velocity error $\dot{e}_q := \dot{q}_s - \dot{q}_m$ asymptotically converges to zero, $\lim_{t\to\infty}\dot{e}_q = 0$.
2) In the free motion case ($\tau_h = \tau_e = 0$), the states get synchronized, $\lim_{t\to\infty} e_q = \lim_{t\to\infty}\dot{e}_q = 0$.
3) If $\dot{q}_i = \ddot{q}_i = 0$, the environmental contact force is accurately transmitted back to the human operator.

Since the essential goal of Section III is to demonstrate a constructive methodology for attack design, assumptions (A1)–(A3) are required so that a PD-like BTOS control scheme can be considered for the theoretical attack design. Attack design for other advanced BTOS control architectures can be similarly accomplished, and assumptions specific to the chosen architecture would then replace (A1)–(A3). The network assumption (A1) is used in this section to simplify the theoretical attack design. It is not assumed in the experiments of Section V, and it is also not explicitly required in the attack detection scheme in Section IV. The consequences of realistic network effects for implementing the proposed detection scheme are discussed in Remark 2.

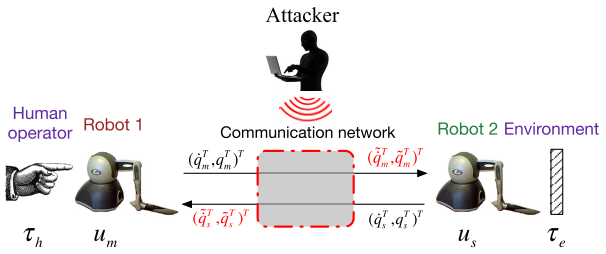

Fig. 2. Attacker can externally compromise the exchanged data between the two robots.

B. Attack Design

First, some assumptions are made for the attacker as follows.

(A4) Attacker has the knowledge of system dynamics structure, controller structure and controller gains.

(A5) Attacker has the ability to receive, interpret, manipulate, and forward the data between the two robots.

(A6) Assume the attack starts at $t = t_a \geq 0$, and $q(t_a) \notin \boldsymbol{E}$ where

$$\boldsymbol{E} = \left\{ q := \left(\dot{q}_s^T, q_s^T, \dot{q}_m^T, q_m^T\right)^T | \dot{q}_s = \dot{q}_m = 0, q_s = q_m \right\} \quad (7)$$

is the equilibrium set for BTOS.

Now, a formal definition of the attacker considered in this brief can be given as follows.

Definition 1: In this brief, the attacker is a malicious external entity with assumptions (A4)–(A6) which can launch an edge attack called false data injection attack by injecting false data $\hat{d} \in \mathbb{R}^p$ into the original data $d \in \mathbb{R}^p$ and thereby modifying d to $\tilde{d} \in \mathbb{R}^p$ as the following mapping g:

$$g : \mathbb{R}^p \to \mathbb{R}^p, \ d \mapsto \tilde{d} = d + \hat{d} \quad (8)$$

where $\hat{d} \in \mathbb{R}^p$ is also referred to as injection data in this brief.

For the BTOS with a PD-like control scheme, as shown in Fig. 2, the attacker can launch false data injection attacks, which can intelligently replace the data content $(\dot{q}_i^T, q_i^T)^T$ being exchanged by the master and slave robots with $(\tilde{q}_i^T, \tilde{q}_i^T)^T$ for $i = m, s$.

The formal definition of DFDIA can be given as follows.

Definition 2: A false data injection attack that injects the false data in the data content transmitted in communication links and ensures that a positive semidefinite system storage function V satisfies $\lim_{t\to\infty} V = \infty$ with $\dot{V} \geq 0$, $\forall t \geq t_a$ is called DFDIA with respect to a storage function V.

In this brief, for BTOS, we study the DFDIA with respect to a specific form of storage function given as

$$V = \frac{1}{2}\dot{q}_s^T M_s \dot{q}_s + \frac{1}{2}\dot{q}_m^T M_m \dot{q}_m + \frac{1}{2}(q_s - q_m)^T K_p (q_s - q_m) - \int_0^t \dot{q}_m^T \tau_h ds + \beta_h + \int_0^t \dot{q}_s^T \tau_e ds + \beta_e \quad (9)$$

where the storage function V in (9) can be used as a Lyapunov-like function for the system (5) to demonstrate master–slave synchronization in the system. Clearly, DFDIA can not only prevent the BTOS synchronization but also lead to violent instability in the overall system. V is positive semidefinite due to the property (P1) and assumption (A2). An increasing V implies that controllers can eventually drive the motors of the revolute robotic manipulators to their maximum speed limits for an indefinite amount of time. This can put stress on the mechanical system of the robots, thereby making the BTOS unsafe for human users and the environment. Hence, it is essential to study the impact of such false data injection attacks. It should be noted that it is not always necessary for the attacker to launch DFDIA if the attacker's goal is just to disrupt the system, but here the DFDIA can be considered as the worst case attack design. The detection scheme for a general false data injection attack is discussed in Section IV.

Theorem 1: Consider a BTOS with dynamics (5) controlled by (6). A false data injection attack that modifies the states being exchanged between the master and the slave robots as

$$\tilde{q} = q + \hat{q} = q + Kq \quad (10)$$

where $q = (\dot{q}_s^T, q_s^T, \dot{q}_m^T, q_m^T)^T$, $\tilde{q} = (\tilde{\dot{q}}_s^T, \tilde{q}_s^T, \tilde{\dot{q}}_m^T, \tilde{q}_m^T)^T$, and "attack gain" constant $K \in \mathbb{R}^{4n\times 4n}$ is a DFDIA with respect to the storage function V in (9) if and only if: 1) $P := A + BK$ is positive semidefinite and 2) the set $\boldsymbol{S}\backslash\boldsymbol{E}$ is not positively invariant, where $\boldsymbol{S} := \{q \in \mathbb{R}^{4n} | q^T P q = 0\}$

$$A = \begin{bmatrix} -(K_{dm}+K_d)I_n & 0_n & K_d I_n & 0_n \\ 0_n & 0_n & 0_n & 0_n \\ K_d I_n & 0_n & -(K_{dm}+K_d)I_n & 0_n \\ 0_n & 0_n & 0_n & 0_n \end{bmatrix}$$

$$B = \begin{bmatrix} 0_n & 0_n & K_d I_n & K_p I_n \\ 0_n & 0_n & 0_n & 0_n \\ K_d I_n & K_p I_n & 0_n & 0_n \\ 0_n & 0_n & 0_n & 0_n \end{bmatrix}.$$

Proof: Consider the storage function given in (9) for the BTOS. Clearly, $V > 0, \forall q \in \mathbb{R}^{4n}\backslash\boldsymbol{E}$.

Using properties (P1) and (P2), the derivative of V along the system trajectories described by (5) and (6) is given by

$$\begin{aligned} \dot{V} &= \frac{1}{2}\dot{q}_s^T \dot{M}_s \dot{q}_s + \dot{q}_s^T M_s \ddot{q}_s + \frac{1}{2}\dot{q}_m^T \dot{M}_m \dot{q}_m + \dot{q}_m^T M_m \ddot{q}_m \\ &\quad + (\dot{q}_s - \dot{q}_m)^T K_p (q_s - q_m) - \dot{q}_m^T \tau_h + \dot{q}_s^T \tau_e \\ &= \dot{q}_s^T u_s + \dot{q}_m^T u_m + (\dot{q}_s - \dot{q}_m)^T K_p (q_s - q_m) \\ &= -\dot{q}_s^T K_{dm} \dot{q}_s - \dot{q}_m^T K_{dm} \dot{q}_m - \dot{q}_s^T (K_d(\dot{q}_s - \tilde{\dot{q}}_m) + K_p(q_s - \tilde{q}_m)) \\ &\quad - \dot{q}_m^T (K_d(\dot{q}_m - \tilde{\dot{q}}_s) + K_p(q_m - \tilde{q}_s)) + (\dot{q}_s - \dot{q}_m)^T K_p (q_s - q_m). \end{aligned} \quad (11)$$

Substituting $\tilde{q}$ from (10) into (11), and rearranging, we get

$$\dot{V} = q^T (A + BK) q = q^T P q. \quad (12)$$

Sufficiency:

From (A6), $q(t_a) \notin \boldsymbol{E}$, which means $V(t_a)$ is positive. Using (12), the following conclusions can be made.

1) If P is positive semidefinite, then V is always positive $\forall t \geq t_a$ and q never enters $\boldsymbol{E}$.
2) If the set $\boldsymbol{S}\backslash\boldsymbol{E}$ is not positively invariant, then $\dot{V}$ does not always stay in 0, thus V ultimately keeps increasing.

Hence, from Definition 2, a false data injection attack given by (10) is DFDIA with respect to the storage function V in (9).

Necessity:

By Definition 2, if the false data injection attack is DFDIA, then $\dot{V} \geq 0$ in (12), which implies P is positive semidefinite. In addition, as $V \to \infty$ as $t \to \infty$, the set $\mathbf{S}\backslash\mathbf{E}$ cannot be positively invariant. Hence, both (1) and (2) are necessary.

■

As the "attack gain" K is a constant matrix, the false data injection attack (10) is referred to as static DFDIA.

Remark 1: Among different choices for the attacker to inject the false data, the proposed DFDIA (10) guarantees the increase of the storage function V in (9) and also gives a provision for controlling the degree of instability in the BTOS by adjusting $\dot{V}$ in (12), and this point can be further demonstrated in the proof of the next proposition.

The following proposition discusses a simple DFDIA design method with the help of Theorem 1.

Proposition 1: Consider a BTOS with dynamics (5) controlled by (6). A false data injection attack that modifies the states being exchanged between the master and slave robots in the form given by (10) is a DFDIA with respect to the storage function V in (9), if the "attack gain"

$$K = \begin{bmatrix} K_{11} & K_{12} & K_{13}+\Lambda_3 & K_{14} \\ K_{21} & K_{22} & K_{23}+\Lambda_4 & K_{24} \\ K_{31}+\Lambda_1 & K_{32} & K_{33} & K_{34} \\ K_{41}+\Lambda_2 & K_{42} & K_{43} & K_{44} \end{bmatrix}$$

where $K_{ij} \in \mathbb{R}^{n\times n}$ are constant coefficient matrices satisfying

$$\begin{aligned} -(K_d+K_{dm})I_n + K_dK_{31} + K_pK_{41} &= 0_n \\ -(K_d+K_{dm})I_n + K_dK_{13} + K_pK_{23} &= 0_n \\ K_dI_n + K_dK_{11} + K_pK_{21} &= 0_n \\ K_dI_n + K_dK_{33} + K_pK_{43} &= 0_n \\ K_dK_{32} + K_pK_{42} = 0_n,\ \ K_dK_{12} + K_pK_{22} &= 0_n \\ K_dK_{34} + K_pK_{44} = 0_n,\ \ K_dK_{14} + K_pK_{24} &= 0_n \end{aligned} \quad (13)$$

and $\{\Lambda_j\} \in \mathbb{R}^{n\times n}$ are positive definite matrices.

Proof: The proof follows by verifying that the sufficient conditions in Theorem 1 can be satisfied with the proposed K. A detailed proof is provided in the Appendix.

■

IV. Attack Detection for BTOS

As shown in Section III, DFDIA is only a special false data injection attack. To just disrupt the system, the attacker may not launch DFDIA but just launch a general false data injection attack defined as (8), which can be arbitrary false data injections on data content. In this section, a physics-based attack detection scheme with an encoding–decoding structure is proposed to detect general false data injection attacks (including DFDIA as a special case).

The proposed scheme is based on the inherent physics of the transmitted data, which in this case means that the physical relationship between the received velocities and positions should be satisfied at all times. This implies that the received data $(\tilde{q}_i, \tilde{\dot{q}}_i)$ are only accepted when the following detection condition is satisfied:

$$\tilde{\dot{q}}_i(t) = \frac{d\tilde{q}_i(t)}{dt}, \quad \forall t \geq 0. \quad (14)$$

It should be noted that the simple checking condition (14) is not adequate for protecting the system because it is easy for the attacker to design the injection data $(\hat{q}_i, \hat{\dot{q}}_i)$ for $(q_i, \dot{q}_i)$ such that the checking condition (14) is satisfied, which means the detection is bypassed. Thus, here a physics-based attack detection scheme with an encoding–decoding structure is proposed based on the simple condition (14) to detect general false data injection attacks. In practice, the scheme is implemented in discrete time. For $i = m, s$, let T_i denote the sufficiently small sampling period for the master and slave robots. On each side, the data are transmitted at the time instance $t = \{k_iT_i | k_i = 1, 2, 3...\}$, where $k_i \in \mathbb{Z}$ is the transmission sequence number.

In the rest of this brief, for the notation such as ${}^i\lambda_{i1}^{k_i}$, the superscript k_i denotes the sequence number k_i, and i in the upper left denotes the transmission direction. In the proposed encoding–decoding scheme, for $i = m$ or s and $j = s$ or m, a set of encoding factors ${}^i\lambda_{i1}, {}^i\lambda_{i2} \in \mathbb{R}$ and decoding factors ${}^i\lambda_{j1}, {}^i\lambda_{j2} \in \mathbb{R}$ are utilized. For each transmission sequence k_i, the encoding and decoding factors are updated with the same dynamics f_1, f_2 as

$$\begin{aligned} {}^i\lambda_{i1}^{k_i} &= {}^i\lambda_{i1}^{k_i-1} + T_i f_1\big({}^i\lambda_{i1}^{k_i-1}, k_i\big) \\ {}^i\lambda_{i2}^{k_i} &= {}^i\lambda_{i2}^{k_i-1} + T_i f_2\big({}^i\lambda_{i2}^{k_i-1}, k_i\big) \end{aligned} \quad (15)$$

and

$$\begin{aligned} {}^i\lambda_{j1}^{k_i} &= {}^i\lambda_{j1}^{k_i-1} + T_i f_1\big({}^i\lambda_{j1}^{k_i-1}, k_i\big) \\ {}^i\lambda_{j2}^{k_i} &= {}^i\lambda_{j2}^{k_i-1} + T_i f_2\big({}^i\lambda_{j2}^{k_i-1}, k_i\big) \end{aligned} \quad (16)$$

where functions $f_1, f_2 : \mathbb{R}\times\mathbb{Z} \to \mathbb{R}$ are known to both master and slave sides and f_1, f_2 need to be designed such that ${}^i\lambda_{j1}^{k_i} \neq {}^i\lambda_{j2}^{k_i}$. If the initial values for $({}^i\lambda_{i1}, {}^i\lambda_{i2})$ and $({}^i\lambda_{j1}, {}^i\lambda_{j2})$ can set to be equal as $({}^i\lambda_{i1}^0, {}^i\lambda_{i2}^0) = ({}^i\lambda_{j1}^0, {}^i\lambda_{j2}^0) = ({}^i\lambda_1^0, {}^i\lambda_2^0)$, then the encoding and decoding factors remain equal as ${}^i\lambda_{i1}^{k_i} = {}^i\lambda_{j1}^{k_i} = {}^i\lambda_1^{k_i}$ and ${}^i\lambda_{i2}^{k_i} = {}^i\lambda_{j2}^{k_i} = {}^i\lambda_2^{k_i}$.

For the proposed detection scheme to work satisfactorily, the following assumption needs to be made.

(A7) For $i = m$ or s, the initial values for the encoding factors $({}^i\lambda_{i1}^0, {}^i\lambda_{i2}^0)$ and decoding factors $({}^i\lambda_{j1}^0, {}^i\lambda_{j2}^0)$ are set to be equal as $({}^i\lambda_1^0, {}^i\lambda_2^0)$. These are shared securely between both sides before the operation starts so that the attacker does not know the value of ${}^i\lambda_1^{k_i}, {}^i\lambda_2^{k_i}$ during the operation.[1]

Under assumption (A7), for the sake of simplicity, only the notations ${}^i\lambda_1^{k_i}, {}^i\lambda_2^{k_i}$ are utized to denote both the encoding and decoding factors. Now, the implementation details can be outlined as the following.

For each transmission sequence $k_i = 1, 2, 3, ...$

1) On the sending end $i = m$ or s, first update the encoding factors ${}^i\lambda_1, {}^i\lambda_2$ as (15). Then, instead of sending

[1]It can be argued that the attacker may deduce the initial values $({}^i\lambda_1^0, {}^i\lambda_2^0)$ using an observer-based approach. Initial investigation to mitigate this possibility has been accomplished in [32].

$(\dot{q}_i^{k_i}, q_i^{k_i})$, send the encoded data $(r_{i1}^{k_i}, r_{i2}^{k_i})$ with $(r_{i1}^{k_i}, r_{i2}^{k_i})$ given as

$$r_{i1}^{k_i} = \dot{q}_i^{k_i} + {}^i\lambda_1^{k_i} q_i^{k_i}, \quad r_{i2}^{k_i} = \dot{q}_i^{k_i} + {}^i\lambda_2^{k_i} q_i^{k_i}. \tag{17}$$

2) On the receiving end $j = s$ or m, after receiving the data, first update the decoding factors ${}^i\lambda_1, {}^i\lambda_2 \in \mathbb{R}$ as (16). Then, recover $\dot{q}_i^{k_i}, q_i^{k_i}$ by decoding $(r_{i1}^{k_i}, r_{i2}^{k_i})$ as

$$q_i^{k_i} = {}^i a^{k_i}\left(r_{i1}^{k_i} - r_{i2}^{k_i}\right), \quad \dot{q}_i^{k_i} = {}^i b^{k_i} r_{i1}^{k_i} - {}^i c^{k_i} r_{i2}^{k_i} \tag{18}$$

where ${}^i a^{k_i} = 1/({}^i\lambda_1^{k_i} - {}^i\lambda_2^{k_i})$, ${}^i b^{k_i} = {}^i\lambda_2^{k_i}/({}^i\lambda_2^{k_i} - {}^i\lambda_1^{k_i})$, ${}^i c^{k_i} = {}^i\lambda_1^{k_i}/({}^i\lambda_2^{k_i} - {}^i\lambda_1^{k_i})$.
Note ${}^i a^{k_i}, {}^i b^{k_i}, {}^i c^{k_i}$ are only valid when ${}^i\lambda_1^{k_i} \neq {}^i\lambda_2^{k_i}$.

3) After decoding, based on (14), check the physical relation between $q_i^{k_i}$ and $\dot{q}_i^{k_i}$ using

$$\frac{q_i^{k_i} - q_i^{k_i-1}}{T_i} = \dot{q}_i^{k_i}. \tag{19}$$

The value q_i^0 should also be shared securely between both sides before the operation starts; thus, the condition (19) can be ready to check when both sides start to sending the data packet from $k_i = 1$.

4) If the detection condition (19) is not violated, utilize $\dot{q}_i^{k_i}, q_i^{k_i}$ in the controller. Otherwise, the attack is detected and rejects the received data to protect the system.

The above-mentioned detection procedure can be summarized as Algorithm 1

Algorithm 1 Attack Detection Algorithm for $i = m, s$

```
for k = 1, 2, 3, ... do
  Update ^iλ1, ^iλ2 as (15) {on the sending end}
  Encode q̇_i^{k_i}, q_i^{k_i} and send (r_i1^{k_i}, r_i2^{k_i}) as (17)
  Update ^iλ1, ^iλ2 as (16) {on the receiving end}
  Decode (r_i1^{k_i}, r_i2^{k_i}) as (18) to recover q̇_i^{k_i}, q_i^{k_i}
  Check detection condition (19)
  if (19) is violated then
    Attack is detected and reject the received data
  else
    Utilize the received q̇_i^{k_i}, q_i^{k_i} in the controller
  end if
end for
```

The following theorem provides a necessary condition for a general false data injection attack to avoid detection.

Theorem 2: With the proposed attack detection scheme, suppose a general false data injection attack defined as (8) is launched at transmission sequence k_i^a and for $k_i \geq k_i^a$, the attacker can modify $(r_{i1}^{k_i}, r_{i2}^{k_i})$ to $(\tilde{r}_{i1}^{k_i}, \tilde{r}_{i2}^{k_i})$ as

$$\tilde{r}_{i1}^{k_i} = r_{i1}^{k_i} + \hat{r}_{i1}^{k_i}, \quad \tilde{r}_{i2}^{k_i} = r_{i2}^{k_i} + \hat{r}_{i2}^{k_i} \tag{20}$$

where $\hat{r}_{i1}^{k_i}, \hat{r}_{i2}^{k_i} \in \mathbb{R}^n$ are the arbitrary injection data. Then, the false data injection attack (20) can avoid the proposed detection scheme only if

$$\left({}^i a^{k_i} - T_i\, {}^i b^{k_i}\right)\hat{r}_{i1}^{k_i} - \left({}^i a^{k_i} - T_i\, {}^i c^{k_i}\right)\hat{r}_{i2}^{k_i} = {}^i a^{k_i-1}\left(\hat{r}_{i1}^{k_i-1} - \hat{r}_{i2}^{k_i-1}\right) \tag{21}$$

where ${}^i a^{k_i}, {}^i b^{k_i}, {}^i c^{k_i}$ are defined in (18) and T_i is the sufficiently small sampling period for $i = m, s$.

Proof: Assume that the false data injection attack (20) has been launched. From (18), the received data are decoded as

$$\tilde{q}_i^{k_i} = {}^i a^{k_i}\left(\tilde{r}_{i1}^{k_i} - \tilde{r}_{i2}^{k_i}\right), \quad \tilde{\dot{q}}_i^{k_i} = {}^i b^{k_i}\tilde{r}_{i1}^{k_i} - {}^i c^{k_i}\tilde{r}_{i2}^{k_i}. \tag{22}$$

Executing the detection condition (19) for $\tilde{q}_i^{k_i}$ and $\tilde{\dot{q}}_i^{k_i}$ from (22), using (18), and substituting $\tilde{q}_i^{k_i}$ from (22) into the left-hand side of (19) gives

$$\frac{\tilde{q}_i^{k_i} - \tilde{q}_i^{k_i-1}}{T_i} = \frac{q_i^{k_i} - q_i^{k_i-1}}{T_i} + \frac{{}^i a^{k_i}\left(\hat{r}_{i1}^{k_i} - \hat{r}_{i2}^{k_i}\right)}{T_i} - \frac{{}^i a^{k_i-1}\left(\hat{r}_{i1}^{k_i-1} - \hat{r}_{i2}^{k_i-1}\right)}{T_i}. \tag{23}$$

Substituting $\tilde{\dot{q}}_i^{k_i}$ from (22) into the right-hand side of (19), we get

$$\tilde{\dot{q}}_i^{k_i} = \dot{q}_i^{k_i} + {}^i b^{k_i}\hat{r}_{i1}^{k_i} - {}^i c^{k_i}\hat{r}_{i2}^{k_i}. \tag{24}$$

Assume the attacker can avoid the detection condition (19), which means $(\tilde{q}_i^{k_i} - \tilde{q}_i^{k_i-1})/T_i = \tilde{\dot{q}}_i^{k_i}$, thus it requires

$$\frac{{}^i a^{k_i}\left(\hat{r}_{i1}^{k_i} - \hat{r}_{i2}^{k_i}\right)}{T_i} - \frac{{}^i a^{k_i-1}\left(\hat{r}_{i1}^{k_i-1} - \hat{r}_{i2}^{k_i-1}\right)}{T_i} = {}^i b^{k_i}\hat{r}_{i1}^{k_i} - {}^i c^{k_i}\hat{r}_{i2}^{k_i} \tag{25}$$

which is equivalent to

$$\left({}^i a^{k_i} - T_i\, {}^i b^{k_i}\right)\hat{r}_{i1}^{k_i} - \left({}^i a^{k_i} - T_i\, {}^i c^{k_i}\right)\hat{r}_{i2}^{k_i} = {}^i a^{k_i-1}\left(\hat{r}_{i1}^{k_i-1} - \hat{r}_{i2}^{k_i-1}\right). \tag{26}$$

Hence, the false data injection attack (20) can avoid the detection only if (26) is satisfied.

■

Condition (26) represents a complex relation between two consecutive injection data. When ${}^i\lambda_1^{k_i}$ and ${}^i\lambda_2^{k_i}$ are unknown to the attacker during the operation, the condition (26) cannot be satisfied; thus, the attack can be detected. The following corollary provides a simpler sufficient condition for the attack detection.

Corollary 1: With the proposed attack detection scheme, suppose the attack (20) is launched at transmission sequence k_i^a. The false data injection attack (20) is detected if

$$\frac{\hat{r}_{i1}^{k_i^a}}{\hat{r}_{i2}^{k_i^a}} \neq \frac{1 + T_i\, {}^i\lambda_1^{k_i^a}}{1 + T_i\, {}^i\lambda_2^{k_i^a}} \tag{27}$$

where ${}^i\lambda_1^{k_i}$ and ${}^i\lambda_2^{k_i}$ are the encoding and decoding factors.

Proof: The false data injection attack (20) is launched at $k_i = k_i^a \geq 1$, then the data content before $k_i = k_i^a$ is not compromised. Thus, $\hat{r}_{i1}^{k_i-1}, \hat{r}_{i2}^{k_i-1}$ equal 0 in condition (26) at $k_i = k_i^a$.

Hence, at $k_i = k_i^a$, the condition (26) is reduced to

$$\frac{\hat{r}_{i1}^{k_i^a}}{\hat{r}_{i2}^{k_i^a}} = \frac{{}^i a^{k_i^a} - T_i\, {}^i c^{k_i^a}}{{}^i a^{k_i^a} - T_i\, {}^i b^{k_i^a}} = \frac{1 + T_i\, {}^i\lambda_1^{k_i^a}}{1 + T_i\, {}^i\lambda_2^{k_i^a}}. \tag{28}$$

Since ${}^i\lambda_1^{k_i}$ and ${}^i\lambda_2^{k_i}$ are unknown to the attacker during the operation, the false data injection attack (20) is detected once the attack is launched at $k_i = k_i^a$ if (28) is not satisfied.

■

Remark 2: As noted earlier, the scheme is solely based on the data between two consecutive sequences k_i and $k_i - 1$. Compared with the detection scheme in [28], first, here the detection scheme is presented in a discrete time setting, which is more realistic in implementation; second, the encoding and decoding factors are not updated synchronously as [28] but are updated asynchronously once the packet is sent or received as (15) and (16); thus, the current scheme is independent of the network delay. Due to other network factors such as noises, packet drops, and quantization errors, the checking condition in (19) can be relaxed as

$$e_i^{k_i} := \left\| \frac{q_i^{k_i} - q_i^{k_i-1}}{T_i} - \dot{q}_i^{k_i} \right\| < \epsilon_i \tag{29}$$

where e_i is termed the checking error and the threshold $\epsilon_i > 0$ can be determined empirically, as described in Section V-B. Another potential method to better handle the packet drops is described next. In each packet on the sending end, the data (r_{i1}, r_{i2}) of past n consecutive sequences can be appended, then even if $m < n$ consecutive packets are dropped, the detection condition (29) can still be checked at the receiving end.

Remark 3: Unlike the model-based attack detection schemes in [16], [17], [21], and [22], the proposed attack detection scheme is solely based on the inherent physical relation within the transmitted data and does not require the knowledge of the system parameters. The detection condition is easy to compute and only relies on two consecutive data samples. Thus, the proposed detection scheme has the following advantages: it is fast, computationally light, and does not require knowledge of system parameters.

V. Experiment

In this section, the theoretical results were tested on a BTOS experimental platform that consists of two PHANToM Omni robots. The PHANToM Omni robot, developed by SensAble Technology, is a six-degree-of-freedom (DOF) haptic device which can apply a force feedback on the user's hand and allow the user to feel virtual objects and interact with the virtual environment. The Omni robot consists of six revolute joints where the last three joints are not actuated. Since the first three joints are actuated, it can be used as a low-cost three-DOF manipulator in the research of robotics control.

To set up a BTOS experimental platform, two PHANToM Omni haptic robots were used. Each robot was interfaced to a machine powered with Intel Core2Quad processor, 8-GB RAM and operating system Windows 7, using FireWire 400 (IEEE 1394) cable. The program to control each robot was written in C++ using the OpenHaptics API (v 3.1), which is developed by SensAble Technologies for PHANToM devices. For nominal BTOS operations, the considered PD-like control (in Section III-A) with empirically fine-tuned gains was implemented.

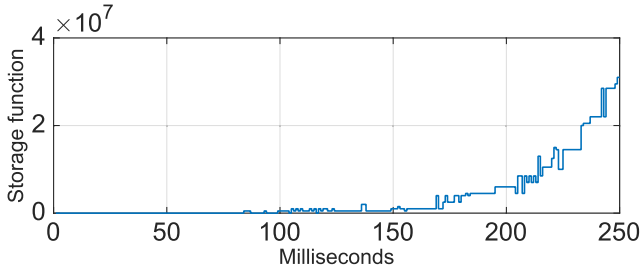

Fig. 3. Storage function V under static DFDIA.

For networking, Windows socket API was utilized and packets were transmitted in user datagram protocol (UDP) format through the Ethernet. The payload of each packet consisted of the robot's joint angles and rate of change of respective joint angles. There was no packet fragmentation required.

The application layer of the program consisted of three supplementary threads along with the primary main thread, which initialized the network sockets, and started and ended the supplementary threads. The first thread was responsible for interacting with the robot, synchronizing with encoders and joint motors, computing the rate of change of joint angles, and the control action for each motor. The second thread was responsible for handling the inbound packets, reading the UDP sockets, parsing the packet payload, and extracting the values of joint angles and their rate of change. The third thread handled the outbound packets, created the packet payload, and queued it at the UDP socket for transmission. The transport layer and layers beneath it were implemented by the Windows kernel.

To emulate the "man-in-the-middle" attacker, equivalent changes to the received packet payload were made in the *Second thread* (receiving thread) of the robots' application programs. The "man-in-the-middle" can also be easily emulated by a third computer relaying between the two robots.

A. Attack Design for DFDIA

Assuming no detection scheme in BTOS, a simple static DFDIA was designed using Proposition 1 to attack the system. The following control gains were selected:

$$K_p = 600, \quad K_d = 5, \text{ and } K_{dm} = 5.$$

The "attack gain" $K \in \mathbb{R}^{12\times 12}$ was designed as

$$K = \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 \\ K_{21} & 0_3 & K_{23} + \Lambda_4 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \\ K_{41} + \Lambda_2 & 0_3 & K_{43} & 0_3 \end{bmatrix}.$$

From (13), $K_{41} = K_{23} = \text{diag}([s_1, s_1, s_1])$, $K_{21} = K_{43} = \text{diag}([s_2, s_2, s_2])$, with $s_1 = (K_d + K_{dm})/K_p$, $s_2 = -K_d/K_p$. Λ_4 and Λ_2 were chosen as $diag([1.5, 1.5, 1.5])$.

In order to protect the system from real damage, an inherent maximum joint speed limit was added on the robots. In addition, the system storage function value was only observed and recorded during a short time interval after launching the attack.

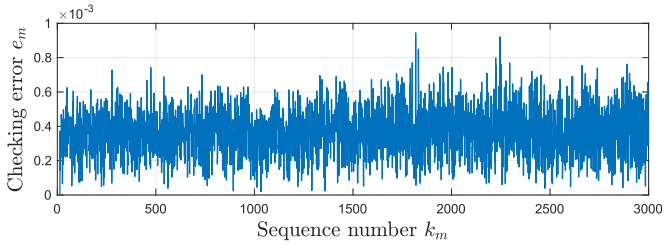

Fig. 4. Checking error e_m on the slave side under a normal operation condition.

Under this attack, the storage function V of BTOS is shown in Fig. 3. The nonsmooth behavior of the storage function was due to the inherent physical limits of the robot joint space.

B. Attack Detection for General False Data Injection Attack

The proposed detection scheme in Section IV was tested under general false data injection attacks. Here, the sampling period $T_i = 4ms$. In the detection scheme, for $i = m, s$, the initial values of $({}^i\lambda_1, {}^i\lambda_2)$ were set to be $({}^i\lambda_1^0, {}^i\lambda_2^0) = (5, -2)$ and the update laws f_1, f_2 in (15) and (16) were designed as

$$f_1 = 20\cos(10k_iT_i), \quad f_2 = -20\sin(10k_iT_i), \quad k_i = 1, 2, 3, \ldots$$

It can be verified that ${}^i\lambda_1^{k_i} \neq {}^i\lambda_2^{k_i}, k_i = 1, 2, 3, \ldots$.

In Case 1, the proposed detection scheme was tested without additional artificial network delay. Then, in the Case 2, in order to verify that the detection scheme is independent of network delay as discussed in Remark 2, a 400-ms one-way network delay was added artificially using a software called clumsy.[2]

Case 1: In order to determine the threshold ϵ_i in (29), the BTOS was run with the detection scheme implemented under normal operation condition (no attack) for a certain period of time. The saved state data were then utilized to compute the checking error e_i in (29), and the value of ϵ_i was set to exceed the computed e_i. In this experiment, to determine ϵ_m, the master robot was utilized to teleoperate the slave robot under a normal operation condition for a time period of $10s$, and the computed checking error on the slave side e_m is shown in Fig. 4. Based on this plot, the threshold was selected as $\epsilon_m = 0.0015$. It is also worth noting that the teleoperation performance remained unaffected during the operation in the absence of any false data injection attack.

In the experiment, the attacker launched a general false data injection attack on $(r_{m1}^{k_m}, r_{m2}^{k_m})$ from $k_m = 3000$. Following (20), the injection data for $(r_{m1}^{k_m}, r_{m2}^{k_m})$ were designed as $(\hat{r}_{m1}^{k_m}, \hat{r}_{m2}^{k_m}) = (-0.01\sin(0.2t), 0.01\cos(0.2t))$. The plot of the checking error e_m on the slave side is shown in Fig. 5. The attack was successfully detected once the attack was launched at $k_m = 3000$ since the checking error e_m instantly exceeded the threshold $\epsilon_m = 0.0015$ (red line in Fig. 5). In this experiment, when checking condition in (29) was violated, in order to protect the system, the robots just dropped the received data and utilized the previously received packet values to compute the control action.

[2]https://jagt.github.io/clumsy/index.html

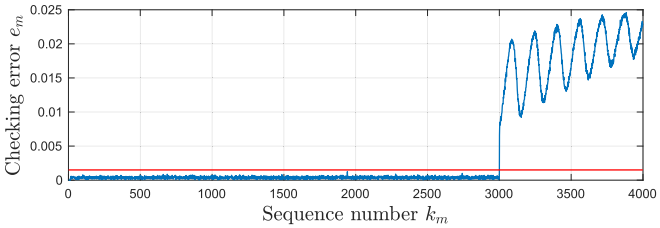

Fig. 5. Checking error e_m (blue line) on the slave side under a general false data injection attack. Red line: determined threshold $\epsilon = 0.0015$.

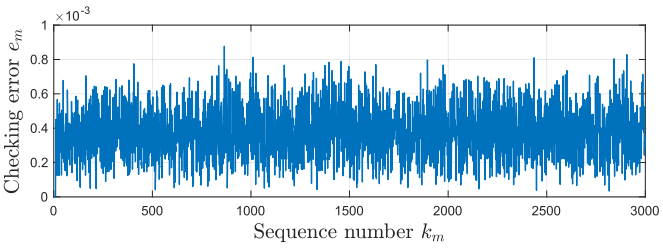

Fig. 6. Checking error e_m on the slave side under a normal operation condition with an artificial network delay.

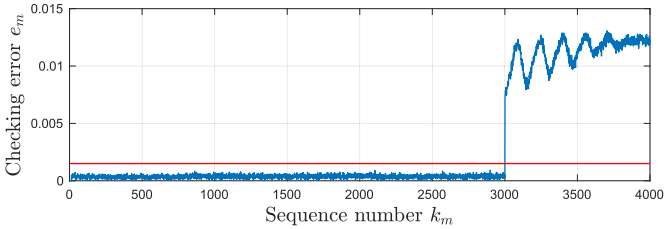

Fig. 7. Checking error e_m (blue line) on the slave side under a general false data injection attack with an artificial network delay. Red line: determined threshold $\epsilon = 0.0\,015$.

Case 2: In this case, a 400-ms one-way delay was added in the network. The same approach in Case 1 was adopted to determine the threshold ϵ_i. The computed error e_m is shown in Fig. 6, and ϵ_m was also set as 0.0015. The attacker also launched the same general false data injection attack on $(r_{ma}^{k_m}, r_{mb}^{k_m})$ from $k_m = 3000$ as Case 1. The plot of the checking error e_m on the slave side is shown in Fig. 7. The attack was successfully detected after the attack was launched at $k_m = 3\,000$, as the checking error e_m exceeded the threshold $\epsilon_m = 0.0\,015$ (red line in Fig. 7).

VI. Conclusion

In this brief, false data injection attacks are studied for a nonlinear BTOS where an adversary can change the data content being communicated between the master and slave robots. The main motivation of the work is to utilize the underlying physics of robotic systems to better understand the attack impact and provide a different perspective on the system security from existing cryptographic tools. Consequently, the static DFDIA is designed to demonstrate the attack impact, and a physics-based attack detection scheme with an encoding-decoding structure is proposed to detect general false data injection attacks. The effectiveness of the attack design and the detection scheme is also verified through experiments. The proposed attack detection scheme can potentially be applied for detecting general false data injection attacks in a variety of networked control/robotic systems.

APPENDIX
PROOF OF PROPOSITION 1

The proof of Proposition 1 can be given as follows.

Proof: Consider the storage function V given by (9) and from (12), $\dot{V} = q^T(A + BK)q$.

Let $K = K^* + \hat{K}$, where

$$K^* = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix}$$

$$\hat{K} = \begin{bmatrix} 0_n & 0_n & \Lambda_3 & 0_n \\ 0_n & 0_n & \Lambda_4 & 0_n \\ \Lambda_1 & 0_n & 0_n & 0_n \\ \Lambda_2 & 0_n & 0_n & 0_n \end{bmatrix}.$$

Substituting K into $\dot{V}$ gives

$$\dot{V} = q^T(A + BK^*)q + q^T B\hat{K}q. \tag{30}$$

Using (13) in (30) gives

$$\begin{aligned} \dot{V} &= q^T B\hat{K}q \\ &= \dot{q}_s^T K_d \Lambda_1 \dot{q}_s + \dot{q}_s^T K_p \Lambda_2 \dot{q}_s + \dot{q}_m^T K_d \Lambda_3 \dot{q}_m + \dot{q}_m^T K_p \Lambda_4 \dot{q}_m. \end{aligned} \tag{31}$$

It is obvious that $\dot{V}$ is positive semidefinite, which implies the condition 1) in Theorem 1 is satisfied.

In this case, $\boldsymbol{S} = \{q \in \mathbb{R}^{4n} | \dot{q}_s = \dot{q}_m = 0\}$ (refer to condition 2) in Theorem 1), thus $\boldsymbol{S}\backslash\boldsymbol{E} = \{q \in \mathbb{R}^{4n} | \dot{q}_s = \dot{q}_m = 0, q_s \neq q_m\}$.

With $\tilde{q} = q + Kq$, the controller (6) becomes

$$\begin{aligned} u_m &= (K_d\Lambda_3 + K_p\Lambda_4)\dot{q}_m - K_p(q_m - q_s) \\ u_s &= (K_d\Lambda_1 + K_p\Lambda_2)\dot{q}_s - K_p(q_s - q_m). \end{aligned} \tag{32}$$

When $\dot{q}_s$ and $\dot{q}_m$ are 0, the nonzero $|q_m - q_s|$ term in control can drive the state away from zero, thereby rendering the velocities nonzero. Thus, $\boldsymbol{S}\backslash\boldsymbol{E}$ is not positively invariant set, which satisfies condition 2) of Theorem 1.

Hence, the false data injection attack given in the proposition is DFDIA with respect to storage function V in (9). As it is mentioned in Remark 1, $\dot{V}$ can be adjusted by appropriately selecting $\{\Lambda_j\}$.

■

REFERENCES

[1] W. Wei and Y. Kui, "Teleoperated manipulator for leak detection of sealed radioactive sources," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2. Apr. 2004, pp. 1682–1687.

[2] W.-K. Yoon *et al.*, "Model-based space robot teleoperation of ETS-VII manipulator," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 602–612, Jun. 2004.

[3] J. Funda and R. P. Paul, "A symbolic teleoperator interface for time-delayed underwater robot manipulation," in *Proc. OCEANS*, Oct. 1991, pp. 1526–1533.

[4] S. Kumar and J. Marescaux, *Telesurgery*. Berlin, Germany: Springer-Verlag, 2008.

[5] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, Dec. 2006.

[6] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 269–281, Apr. 2006.

[7] N. Chopra, M. W. Spong, and R. Lozano, "Synchronization of bilateral teleoperators with time delay," *Automatica*, vol. 44, no. 8, pp. 2142–2148, 2008.

[8] E. Nuño, L. Basañez, R. Ortega, and M. Spong, "Position tracking for non-linear teleoperators with variable time delay," *Int. J. Robot. Res.*, vol. 28, no. 7, pp. 895–910, 2009.

[9] J.-H. Ryu, J. Artigas, and C. Preusche, "A passive bilateral control scheme for a teleoperator with time-varying communication delay," *Mechatronics*, vol. 20, no. 7, pp. 812–823, 2010.

[10] H.-C. Hu and Y.-C. Liu, "Passivity-based control framework for task-space bilateral teleoperation with parametric uncertainty over unreliable networks," *ISA Trans.*, vol. 70, pp. 187–199, Sep. 2017.

[11] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White Paper, Symantec Corp., Secur. Response*, vol. 5, no. 6, p. 29, 2011.

[12] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen, "Cyber security of water SCADA systems—Part I: Analysis and experimentation of stealthy deception attacks," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 5, pp. 1963–1970, Sep. 2013.

[13] S. Gorman, "Electricity grid in us penetrated by spies," *Wall Street J.*, vol. 8, pp. 1–3, Apr. 2009.

[14] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography," in *Proc. NIST DRAFT NISTIR*, vol. 8114, Aug. 2016, pp. 1–31.

[15] H. Sandberg, S. Amin, and K. Johansson, "Cyberphysical security in networked control systems: An introduction to the issue," *IEEE Control Syst.*, vol. 35, no. 1, pp. 20–23, Feb. 2015.

[16] F. Pasqualetti, F. Dorfler, and F. Bullo, "Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems," *IEEE Control Syst.*, vol. 35, no. 1, pp. 110–127, Feb. 2015.

[17] E. Eyisi and X. Koutsoukos, "Energy-based attack detection in networked control systems," in *Proc. 3rd Int. Conf. High Confidence Networked Syst.* New York, NY, USA: ACM, 2014, pp. 115–124.

[18] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1495–1508, Jul. 2011.

[19] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.

[20] Z. Feng and G. Hu, "Distributed tracking control for multi-agent systems under two types of attacks," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 5790–5795, 2014.

[21] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.

[22] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Distributed fault detection and isolation resilient to network model uncertainties," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2024–2037, Nov. 2014.

[23] T. Bonaci and H. J. Chizeck, "Surgical telerobotics meets information security," in *Proc. 21st Usenix Security Symp.*, 2012, pp. 1–2.

[24] T. Bonaci, J. Herron, T. Yusuf, J. Yan, T. Kohno, and H. J. Chizeck. (2015). "To make a robot secure: An experimental analysis of cyber security threats against teleoperated surgical robots." [Online]. Available: https://arxiv.org/abs/1504.04339

[25] G. S. Lee and B. Thuraisingham, "Cyberphysical systems security applied to telesurgical robotics," *Comput. Standards Inter.*, vol. 34, no. 1, pp. 225–229, 2012.

[26] M. E. Tozal, Y. Wang, E. Al-Shaer, K. Sarac, B. Thuraisingham, and B.-T. Chu, "On secure and resilient telesurgery communications over unreliable networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2011, pp. 714–719.

[27] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *Proc. 1st Workshop Secure Control Syst.*, 2010, pp. 1–6.

[28] Y. Dong, N. Gupta, and N. Chopra, "Content modification attacks on consensus seeking multi-agent system with double-integrator dynamics," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 26, no. 11, 2016, Art. no. 116305.

[29] Y. Dong, N. Gupta, and N. Chopra, "On content modification attacks in bilateral teleoperation systems," in *Proc. Amer. Control Conf.*, Jul. 2016, pp. 316–321.

[30] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, vol. 3. New York, NY, USA: Wiley, 2006.

[31] T. Hatanaka, N. Chopra, M. Fujita, and M. W. Spong, *Passivity-Based Control and Estimation in Networked Robotics* (Communications and Control Engineering). Cham, Switzerland: Springer, 2015.

[32] Y. Dong and N. Chopra, "Observability-based secure state encryption design for cyberphysical systems," in *Proc. Indian Control Conf. (ICC)*, Jan. 2018, pp. 24–29.