Data-Driven Methods for Stealthy Attacks on TCP/IP-Based Networked Control Systems Equipped With Attack Detectors

Jun-Sheng Wang and Guang-Hong Yang, *Senior Member, IEEE*

***Abstract*—Most of the existing stealthy attack schemes for cyber-physical systems (CPSs) are presented under the assumption that the model parameters of CPS are known to attackers. Presently, there are only a few model-independent stealthy attack approaches, which, however, need the assumption that attackers know sensor measurements and can modify them. This paper aims to remove the aforementioned conservative assumptions and give a stealthy attack methodology for closed-loop CPS with reference signals, that is, transmission control protocol/Internet protocol (TCP/IP)-based networked control systems. To this end, under the condition that the model parameters of the CPS are unknown, a benchmark platform (consisting of an attack detector and a TCP/IP-based networked dc servo system) used for testing the stealthy attack technology is constructed via data-driven methods. A plan is made, which is utilized for eavesdropping the information of the TCP/IP-based CPS. On this basis, an approach to blocking network communications and injecting the false sensor data into the CPS is explored. A closed-loop recursive identification strategy for the dynamic characteristic matrix of the CPS is designed. By employing all of the above-obtained results, a data-driven stealthy attack scheme for the CPS is proposed and, subsequently, its effectiveness and practicability are validated by experiment.**

***Index Terms*—Cyber-physical systems (CPSs), data-driven methods, dc servo systems, stealthy attack, transmission control protocol/Internet protocol (TCP/IP).**

I. Introduction

DURING recent years, the issue on the security of cyber-physical systems (CPSs) has stirred great interest because CPS are at the core of extremely important industrial systems and infrastructures, such as nuclear power stations and smart grids. The security studies of CPS focus on the two aspects, i.e., attack and defense techniques [1], [3]. This paper devotes its attention to the first aspect. The research hotspots of attack technology mainly involve the eavesdropping, denial-of-service (DoS), and stealthy attacks [2]. The eavesdropping attacks only steal the privacy of CPS (e.g., measurement data) and do not inject any information into CPS (i.e., do not affect the operation of CPS at all), so that it is very difficult to detect such attacks. Although not getting the above privacy, the DoS attacks can prevent both actuator data and measurement information from reaching their respective destination, and therefore are devastating for the normal operation of CPS and easily found. The stealthy attacks possess the merits of eavesdropping and DoS attacks, that is, stealthy attacks destroy not only CPS but also remain undetected by attack detectors [4]. Hence, the stealthy attacks are more dangerous for CPS.

Up to now, the study of stealthy attacks has delivered some extremely valuable results [3]–[16]. Nevertheless, these results often depend on the following three assumptions.

Assumption 1 [3]–[12]: Full model knowledge of CPS regarded as attack targets is known to attackers.

Assumption 2 [12]–[16]: Attackers know sensor measurements in CPS.

Assumption 3 [3]–[16]: Attackers can falsify sensor readings in CPS.

However, it is usually difficult for attackers to let the above assumptions hold true in real CPS. Thus, Assumptions 1–3 could limit the practicability of the aforementioned stealthy attack methods.

The communications between various computers on a network are achieved through protocol suites, among which the most widely used and most widely available protocol suite is the transmission control protocol/Internet protocol (TCP/IP). Due to the fact that the TCP/IP provides reliability of data transport using error control, and with the increasing network speed over the Ethernet, the technology of industrial Ethernet networks (e.g., EtherNet/IP, which is a well-known TCP/IP-based application layer protocol for industrial automation) is gradually applied to the field of automatic control. As a result, lots of TCP/IP-based networked control systems (i.e., CPS) emerged in the industrial domain [1], [30]. A TCP/IP-based networked dc servo system built in the present study is a typical TCP/IP-based CPS.

The data-driven methods first arose in the computer science and then were introduced into the control community. Under the lack of accurate explicit models of the controlled plants, the data-driven methods can take advantage of the online or

Manuscript received January 10, 2018; revised April 3, 2018; accepted May 12, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61703429, Grant 61621004, and Grant 61420106016, and in part by the Research Fund of State Key Laboratory of Synthetical Automation for Process Industries under Grant 2013ZCX01. This paper was recommended by Associate Editor G.-P. Liu. *(Corresponding author: Guang-Hong Yang.)*

The authors are with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China, and also with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: wangjs82@126.com; yangguanghong@ise.neu.edu.cn).

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/TCYB.2018.2837874

offline data of such plants to directly devise controllers, evaluate performance, predict and assess system states, or diagnose and accommodate failures [17]–[29].

This paper aims to remove the aforesaid three conservative assumptions and further give a stealthy attack approach for general CPS, i.e., the closed-loop CPS with reference signals. To this end, under the condition that the model parameters of the CPS are unknown, a benchmark platform (consisting of an attack detector and a TCP/IP-based networked dc servo control system) used for the verification of the stealthy attack technology is first constructed via data-driven methods. Then, a plan is made, which is utilized for eavesdropping the information of the CPS built by employing Ethernet hubs. On this basis, an approach to blocking network communications and injecting the false sensor data into CPS is explored. After that, a closed-loop recursive identification strategy for the dynamic characteristic matrix of the CPS is designed. Finally, with the aid of all the above-obtained results, a data-driven stealthy attack scheme for the CPS is proposed, and subsequently, its effectiveness and practicability are validated by experiment.

This paper makes the underlying contributions.

1) Most of the existing stealthy attack methods (such as [3]–[12]) are presented under the precondition that the parameters of the explicit models of CPS are known to attackers. In contrast, the precondition is not required by this paper via both the proposed data-driven recursive identification algorithm and eavesdropping technique.
2) At present, there are only a few model-independent stealthy attack approaches [13]–[16], which, however, need the assumption that attackers know sensor readings and can falsify them. The assumption is removed in this paper by discovering and then using a potential security flaw in TCP/IP-based CPS.

II. Construction of the Benchmark Platform for Verification of Stealthy Attack Technology

On the basis of the dc motor control system developed in [19] and [20], its networked version has been studied. The resulting networked dc servo system (namely the hardware platform for testing stealthy attack methods) is displayed in Fig. 1 and mainly contains the following components: server and client computers, a dc motor, a motor driver (consisting of a silicon controlled rectifier module EUV-25A-II and a bridge rectifier), a data acquisition card PCI-1711U, a load cell, an opto-electronic encoder utilized for measuring the motor velocity, and an electromagnetic brake regarded as a motor load. In addition, a laptop is used as an attacker's computer here. Fig. 2 shows the connections among these components. The rated voltage and velocity of the dc motor are 220 VDC and 1500 rpm, respectively. More information about the motor and its driver as well as the load cell and the encoder can be found in [19] and [20].

Both the devised speed servo controller and attack detector run on the server computer. The client computer sends measurement data to the server computer via a local area network (LAN) and subsequently receives control inputs from the server computer. The stealthy attack program developed on the laptop is to achieve the following goals.

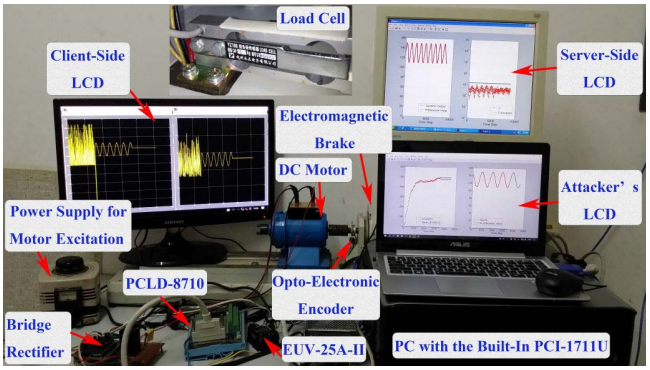

Fig. 1. Benchmark platform for the verification of stealthy attack technology.

1) To eavesdrop communications between the client and server computers.
2) To identify the dynamic characteristic matrix of the networked dc servo system by means of the eavesdropped data.
3) To simulate via the obtained dynamic characteristic matrix, the dc servo motor's behavior under the control inputs calculated by the server computer.
4) To send those simulated outputs back to the server computer so that the attack goes undetected by the attack detector and the communications between the client and server computers are interrupted.

Remark 1: In the following, to show the effect of the proposed stealthy attack plan, the client computer will record both the motor speed and the control command, as can be seen from the client-side liquid crystal display in Fig. 1. Furthermore, for the purpose of convenience, a desktop personal computer of excellent performance serves as the client computer in this investigation. Nevertheless, in practice, a mini-size embedded hardware, such as DSP, ARM, or FPGA, is usually used as the client computer, which is only responsible for sending the sensor information and receiving the control instruction through the LAN, instead of storing or displaying them due to its limited computing and storage capabilities. As a result, the complicated tasks are assigned to a high-performance server computer. Therefore, it is supposed in this paper that only the server computer depicted in Fig. 2 can be utilized for realizing the digital algorithms for a speed servo controller and an attack detector.

The existing design methods for attack detectors depended on the complete information of explicit models of CPS. Unfortunately, the mathematical model of the networked dc servo system constructed in this paper is unknown. Therefore, the data-driven design of attack detectors will be investigated in the following section.

A. Data-Driven Design of Attack Detectors

First of all, the control input, measurement value of a load (i.e., reading of the load cell), rotating speed (i.e., reading of the opto-electronic encoder), and armature current at the

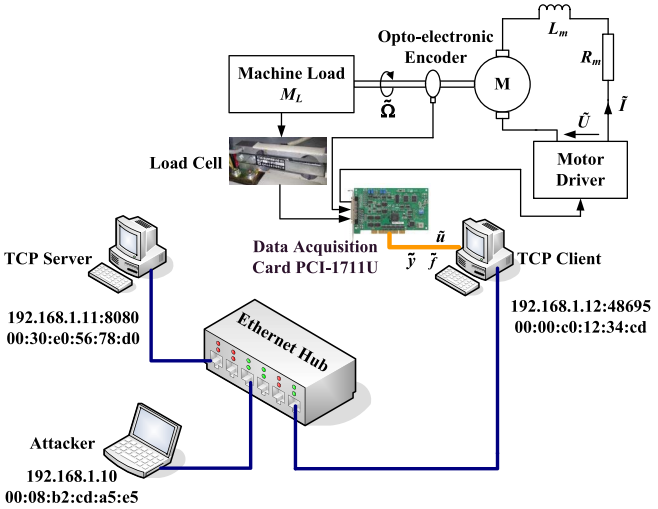

Fig. 2. Block diagram of the benchmark platform for the verification of stealthy attack technology.

time when the motor runs at a certain steady state are, respectively, denoted by $\bar{u}$, $\bar{f}$, $\bar{\Omega}$, and $\bar{I}$. Then, their counterparts at the current instant t are denoted by $\tilde{u}(t)$, $\tilde{f}(t)$, $\tilde{\Omega}(t)$, and $\tilde{I}(t)$, respectively. As a result, to make the methods proposed in this paper suitable for use in the presence of load fluctuations, the dynamics model of the motor system drawn in Fig. 2 is established as

$$\dot{\mathbf{x}}(t) = \mathbf{A_c}\mathbf{x}(t) + \mathbf{B_c}u(t) + \mathbf{E_c}\,f(t) + \mathbf{K_c}w(t) \tag{1}$$

$$y(t) = \mathbf{C_c}\mathbf{x}(t) \tag{2}$$

$$\tilde{y}(t) = y(t) + \bar{y} \tag{3}$$

where

$$\mathbf{A_c} = \begin{bmatrix} -R_m/L_m & -C_{V/\Omega}/L_m \\ C_{T/I}/J & 0 \end{bmatrix},\ \mathbf{B_c} = \begin{bmatrix} K_u/L_m \\ 0 \end{bmatrix}$$

$$\mathbf{C_c} = \begin{bmatrix} 0 & K_y \end{bmatrix},\ \mathbf{E_c} = \begin{bmatrix} 0 \\ -K_f/J \end{bmatrix},\ \mathbf{x}(t) = \begin{bmatrix} I(t) \\ \Omega(t) \end{bmatrix}.$$

$f(t) = \tilde{f}(t) - \bar{f}$, $u(t) = \tilde{u}(t) - \bar{u}$, $\Omega(t) = \tilde{\Omega}(t) - \bar{\Omega}$, $I(t) = \tilde{I}(t) - \bar{I}$, $K_f = M_L(t)/\tilde{f}(t)$, $K_u = \tilde{U}(t)/\tilde{u}(t)$, $K_y = \tilde{y}(t)/\tilde{\Omega}(t)$, $\bar{y} = K_y\bar{\Omega}$, and $M_L(t)$, $C_{T/I}$, $C_{V/\Omega}$, R_m, L_m, $\tilde{U}(t)$, J, and $w(t)$ represent the load torque, motor constant, voltage constant, armature resistance, armature inductance, terminal voltage, total inertia, and external disturbance, respectively. Here, the sampling time T is set to 0.2 s, so that, according to 60 pulses/r, the resolution of the opto-electronic encoder, K_y is equal to 0.2 pulses/rpm, exclusive of which, the other parameters in (1)–(3) are unknown.

Note that, since the digital-to-analog converter built in the PCI-1711U has a resolution of 12 bits, its input $\tilde{u}(t)$ (i.e., the input of the hardware platform displayed in Fig. 1) is actually a dimensionless number between 0 and 4095.

Remark 2: Due to the nonlinear input–output characteristic of the motor driver, the motor terminal voltage $\tilde{U}(t)$ is originally a nonlinear function of $\tilde{u}(t)$, i.e., the input of the digital-to-analog converter of the PCI-1711U. However, by means of (1)–(3), and in light of the methodology given in [20], both load variations and nonlinearity can be compensated by taking advantage of the look-up tables obtained by experiment, so that K_u becomes a constant and the mean value of $f(t)$ is zero. Refer to [20] for more details, which have been neglected in this paper such that we are able to concentrate on the system characteristics derived from network communications. Thus, in the rest of this paper, the nonlinearity of the system is assumed to have been compensated well. Furthermore, without loss of generality, it is supposed that the motor operates under the condition that the electromagnetic brake regarded as a machine load in Fig. 1 is not powered up so as to engender $\bar{f} = 0$. Moreover, let $\bar{u} = 0$. As a result, it is known from the load compensation mechanism reported in [20] that, in the aforementioned case, $\bar{\Omega} = -569.658$ rpm. In summary, the above-obtained values of $\bar{\Omega}$ and $\bar{u}$ are employed in the following experiments.

Since the networked dc servo system under consideration is built up via a 100Mb/s LAN, whose network environment is much better than that of wide area networks, and bandwidth is abundant, there exists very low probability of collisions between the data transmitted by the different nodes in the network. In addition, the TCP/IP can guarantee that the lost packets are retransmitted automatically. As a result, both the packet-loss issue and the problem of time delay in the communication channel are neglected in this paper.

However, the response time for sending and receiving data through the TCP/IP socket softwares of the MATLAB under Windows operating systems is relatively large and therefore causes the control delay, which needs to be coped with.

Consider the networked control system in Fig. 2. Denote the time instant (when the client computer sends a data packet with sensor information to the server computer) as t_1. Once the aforementioned data packet is received by the server computer, it immediately transmits a packet including the corresponding control command to the client computer. If the time instant when the above control command arrives at the client computer is denoted by t_2, then the control delay ς is defined as $\varsigma = t_2 - t_1$.

For the sake of the discretization of (1)–(3) in the form of a linear time-invariant (LTI) model, the maximum control delay of the built networked motor system needs to be ascertained. To this end, a test process has been implemented. Given the selected sampling time, during the test, the MATLAB program running on the client computer sent a packet to the server computer every 0.2 s and recorded the time instant when packets were sent or received. Moreover, the process lasted one hour to get the multiple time-delay values. Finally, the maximum one, 0.063 s, among these values was chosen as the maximum control delay ς_m.

According to ς_m, the delay δ is set to 0.1 s, so that the above networked control system with time-varying control delays can be converted into that with the fixed control delay by resorting to the following method: the client computer does not update the control policy via the input buffer of the TCP client until $t_1 + \delta$, when the control law corresponding to the sensor data at time instant t_1 is bound to reach the client software. Note that the principle used for selecting δ is $\varsigma_m < \delta < T$. On this

basis, a discrete LTI model of (1)–(3) is obtained as

$$\begin{aligned}\mathbf{x}[k+1] &= \mathbf{A_d}\mathbf{x}[k] + \mathbf{B_{d1}}\mathbf{u}[k-1] + \mathbf{B_{d2}}\mathbf{u}[k] \\ &\quad + \mathbf{E_d}\mathbf{f}[k] + \mathbf{K_d}\mathbf{w}[k] \quad (4)\\ \mathbf{y}[k] &= \mathbf{C}_d\mathbf{x}[k] \quad (5)\\ \tilde{\mathbf{y}}[k] &= \mathbf{y}[k] + \bar{y} \quad (6)\end{aligned}$$

where

$$\mathbf{A_d} = e^{\mathbf{A_c}T}, \mathbf{B_{d1}} = \int_{T-\delta}^{T} e^{\mathbf{A_c}\tau}\mathbf{B_c}d\tau, \mathbf{B_{d2}} = \int_{0}^{T-\delta} e^{\mathbf{A_c}\tau}\mathbf{B_c}d\tau$$

$$\mathbf{C_d} = \mathbf{C_c}, \mathbf{E_d} = \int_{0}^{T} e^{\mathbf{A_c}\tau}\mathbf{E_c}d\tau, \mathbf{K_d} = \int_{0}^{T} e^{\mathbf{A_c}\tau}\mathbf{K_c}d\tau$$

$\tau = (k+1)T - t$, $\mathbf{x}[k] = \mathbf{x}(kT)$, and $\mathbf{u}[k]$, $\mathbf{f}[k]$, $\mathbf{w}[k]$, $\mathbf{y}[k]$, and $\tilde{\mathbf{y}}[k]$ have the similar definition to $\mathbf{x}[k]$.

To make the method proposed in this paper applicable to the multiple-input–multiple-output plants, the dimensions of $\mathbf{x}[k]$, $\mathbf{y}[k]$, and $\mathbf{u}[k]$ are, respectively, denoted by n, m, and l in the remaining part of this paper.

Furthermore, in order to provide a uniform and compact representation of symbolic expressions, the underlying notations are introduced.

For the column vector $\boldsymbol{\pi}[k] \in \mathbb{R}^{\varrho}$ that can be $\mathbf{f}[k]$, $\mathbf{s}[k]$, $\mathbf{u}[k]$, $\mathbf{w}[k]$, $\mathbf{x}[k]$, or $\mathbf{y}[k]$, we define

$$\boldsymbol{\pi}_\tau(k) = \begin{bmatrix} \boldsymbol{\pi}[k-\tau+1] \\ \boldsymbol{\pi}[k-\tau+2] \\ \vdots \\ \boldsymbol{\pi}[k] \end{bmatrix} \in \mathbb{R}^{\tau\varrho}$$

$$\boldsymbol{\Pi}_{k,\tau,q} = \Big[\left(\sqrt{\mu}\right)^{q-1}\boldsymbol{\pi}_\tau(k-q+1) \quad \cdots \quad \sqrt{\mu}\boldsymbol{\pi}_\tau(k-1) \quad \boldsymbol{\pi}_\tau(k) \Big] \in \mathbb{R}^{\tau\varrho\times q}$$

where $\mathbf{s}[k]$ stands for the reference signal (i.e., the desired motor speed), τ and q are positive integers, and $0 \ll \mu < 1$ is named a forgetting factor, which makes the old data be down-weighted.

As a result, by (4) and (5), an I/O model is constructed as

$$\mathbf{Y}_{k,f,\alpha} = \mathcal{O}_f\mathbf{X}_{k-f+1,1,\alpha} + \mathcal{H}_f\mathbf{U}_{k-1,f,\alpha} + \boldsymbol{\Upsilon}_f\mathbf{F}_{k,f,\alpha} + \mathcal{G}_f\mathbf{W}_{k,f,\alpha} \quad (7)$$

where f and p represent, respectively, the dimensions of future and past windows, $p \geq f > n$, $\alpha = k-f-p$, $\mathfrak{B} = \mathbf{A_d}\mathbf{B_{d2}} + \mathbf{B_{d1}}$

$$\mathcal{O}_f = \begin{bmatrix} \mathbf{C_d}^T & (\mathbf{C_d}\mathbf{A_d})^T & \cdots & \left(\mathbf{C_d}\mathbf{A_d}^{f-1}\right)^T \end{bmatrix}^T$$

$$\mathcal{H}_f = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \mathbf{C_d}\mathbf{B_{d1}} & \mathbf{C_d}\mathfrak{B} & \mathbf{C_d}\mathbf{B}_{d2} & \ddots & \vdots \\ \mathbf{C_d}\mathbf{A_d}\mathbf{B_{d1}} & \mathbf{C_d}\mathfrak{B} & \mathbf{C_d}\mathbf{B}_{d2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \mathbf{C_d}\mathbf{A_d}^{f-2}\mathbf{B_{d1}} & \mathbf{C_d}\mathbf{A_d}^{f-3}\mathfrak{B} & \cdots & \mathbf{C_d}\mathfrak{B} & \mathbf{C_d}\mathbf{B_{d2}} \end{bmatrix}$$

$$\mathcal{G}_f = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \mathbf{C_d}\mathbf{K_d} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mathbf{C_d}\mathbf{A_d}^{f-2}\mathbf{K_d} & \cdots & \mathbf{C_d}\mathbf{K_d} & 0 \end{bmatrix}$$

and

$$\boldsymbol{\Upsilon}_f = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \mathbf{C_d}\mathbf{E_d} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mathbf{C_d}\mathbf{A_d}^{f-2}\mathbf{E_d} & \cdots & \mathbf{C_d}\mathbf{E_d} & 0 \end{bmatrix}.$$

After that, premultiplying the parity space $\mathcal{O}_f^{\perp}$ [25] on both sides of (7), and moving the input term to the left-hand side (LHS), one has

$$\begin{bmatrix} -\mathcal{O}_f^{\perp}\mathcal{H}_f & \mathcal{O}_f^{\perp} \end{bmatrix}\begin{bmatrix} \mathbf{U}_{k-1,f,\alpha} \\ \mathbf{Y}_{k,f,\alpha} \end{bmatrix} = \mathcal{O}_f^{\perp}\left(\boldsymbol{\Upsilon}_f\mathbf{F}_{k,f,\alpha} + \mathcal{G}_f\mathbf{W}_{k,f,\alpha}\right). \quad (8)$$

Besides, a block Toeplitz matrix is described by

$$\mathscr{H}_f = \begin{bmatrix} 0_{m\times 2l} & 0_{m\times 2l} & \cdots & \cdots & 0_{m\times 2l} \\ \mathbf{C_d}\mathscr{B} & 0_{m\times 2l} & \ddots & \ddots & \vdots \\ \mathbf{C_d}\mathbf{A_d}\mathscr{B} & \mathbf{C_d}\mathscr{B} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{C_d}\mathbf{A_d}^{f-2}\mathscr{B} & \cdots & \mathbf{C_d}\mathbf{A_d}\mathscr{B} & \mathbf{C_d}\mathscr{B} & 0_{m\times 2l} \end{bmatrix} \quad (9)$$

with $\mathscr{B} = \begin{bmatrix} \mathbf{B_{d1}} & \mathbf{B_{d2}} \end{bmatrix}$.

$[\mathbf{S}_{k,f,\alpha}^T, \mathbf{U}_{k-f-1,p,\alpha}^T, \mathbf{Y}_{k-f,p,\alpha}^T]^T$ is first denoted by $\boldsymbol{\Psi}$. Second, applying the QR decomposition to $\mathscr{Z} = [\boldsymbol{\Psi}^T, \mathbf{U}_{k-1,f,\alpha}^T, \mathbf{Y}_{k,f,\alpha}^T]^T$ yields

$$\mathscr{Z} = \begin{bmatrix} \mathbf{R}_{11} & 0 & 0 & 0 & 0 \\ \mathbf{R}_{21} & \mathbf{R}_{22} & 0 & 0 & 0 \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} & 0 & 0 \\ \mathbf{R}_{41} & \mathbf{R}_{42} & \mathbf{R}_{43} & \mathbf{R}_{44} & 0 \\ \mathbf{R}_{51} & \mathbf{R}_{52} & \mathbf{R}_{53} & \mathbf{R}_{54} & \mathbf{R}_{55} \end{bmatrix}\begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \\ \mathbf{Q}_5 \end{bmatrix} \quad (10)$$

which implies

$$\boldsymbol{\Upsilon}_f\mathbf{F}_{k,f,\alpha} + \mathcal{G}_f\mathbf{W}_{k,f,\alpha} = \mathbf{R}_{55}\mathbf{Q}_5 \quad (11)$$

and

$$\boldsymbol{\Psi} = \begin{bmatrix} \mathbf{R}_{11} & 0 & 0 \\ \mathbf{R}_{21} & \mathbf{R}_{22} & 0 \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} \end{bmatrix}\begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \end{bmatrix}. \quad (12)$$

Subsequently, substituting (11) into (8), and then post-multiplying the resulting equation with $\boldsymbol{\Psi}^T$, we can attain

$$\begin{bmatrix} -\mathcal{O}_f^{\perp}\mathcal{H}_f & \mathcal{O}_f^{\perp} \end{bmatrix}\begin{bmatrix} \mathbf{R}_{41} & \mathbf{R}_{42} & \mathbf{R}_{43} \\ \mathbf{R}_{51} & \mathbf{R}_{52} & \mathbf{R}_{53} \end{bmatrix} \times \begin{bmatrix} \mathbf{R}_{11} & 0 & 0 \\ \mathbf{R}_{21} & \mathbf{R}_{22} & 0 \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} \end{bmatrix}^T = 0 \quad (13)$$

in terms of (10). In addition, if the QR decomposition of $[\mathbf{S}_{k-1,f,\alpha-1}^T, \mathbf{U}_{k-f-2,p,\alpha-1}^T, \mathbf{Y}_{k-f-1,p,\alpha-1}^T, \mathbf{U}_{k-2,f,\alpha-1}^T, \mathbf{Y}_{k-1,f,\alpha-1}^T]^T$ at time instant $k-1$ is known, then the corresponding QR factorization at time instant k can be derived

by virtue of

$$\begin{bmatrix} \sqrt{\mu}\mathbf{R}_{11}(k-1) & 0 & 0 \\ \sqrt{\mu}\mathbf{R}_{21}(k-1) & \sqrt{\mu}\mathbf{R}_{22}(k-1) & 0 \\ \sqrt{\mu}\mathbf{R}_{31}(k-1) & \sqrt{\mu}\mathbf{R}_{32}(k-1) & \sqrt{\mu}\mathbf{R}_{33}(k-1) \\ \sqrt{\mu}\mathbf{R}_{41}(k-1) & \sqrt{\mu}\mathbf{R}_{42}(k-1) & \sqrt{\mu}\mathbf{R}_{43}(k-1) \\ \sqrt{\mu}\mathbf{R}_{51}(k-1) & \sqrt{\mu}\mathbf{R}_{52}(k-1) & \sqrt{\mu}\mathbf{R}_{53}(k-1) \end{bmatrix} \begin{bmatrix} 0 & 0 & \mathbf{s}_f(k) \\ 0 & 0 & \mathbf{u}_p(k-f-1) \\ 0 & 0 & \mathbf{y}_p(k-f) \\ \sqrt{\mu}\mathbf{R}_{44}(k-1) & 0 & \mathbf{u}_f(k-1) \\ \sqrt{\mu}\mathbf{R}_{54}(k-1) & \sqrt{\mu}\mathbf{R}_{55}(k-1) & \mathbf{y}_f(k) \end{bmatrix} \boldsymbol{\Phi}(k)$$

$$= \begin{bmatrix} \mathbf{R}_{11}(k) & 0 & 0 \\ \mathbf{R}_{21}(k) & \mathbf{R}_{22}(k) & 0 \\ \mathbf{R}_{31}(k) & \mathbf{R}_{32}(k) & \mathbf{R}_{33}(k) \\ \mathbf{R}_{41}(k) & \mathbf{R}_{42}(k) & \mathbf{R}_{43}(k) \\ \mathbf{R}_{51}(k) & \mathbf{R}_{52}(k) & \mathbf{R}_{53}(k) \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \mathbf{R}_{44}(k) & 0 & 0 \\ \mathbf{R}_{54}(k) & \sqrt{\mu}\mathbf{R}_{55}(k-1) & \check{\mathbf{y}}_f(k) \end{bmatrix} \tag{14}$$

with a proper sequence of orthogonal Givens rotations $\boldsymbol{\Phi}(k)$ [31]. From (13) and (14), it is observed that $[-\mathcal{O}_f^{\perp}\mathcal{H}_f \quad \mathcal{O}_f^{\perp}]$ is in the left null space of

$$\mathcal{Q} = \begin{bmatrix} \mathbf{R}_{41}(k) & \mathbf{R}_{42}(k) & \mathbf{R}_{43}(k) \\ \mathbf{R}_{51}(k) & \mathbf{R}_{52}(k) & \mathbf{R}_{53}(k) \end{bmatrix} \times \begin{bmatrix} \mathbf{R}_{11}(k) & 0 & 0 \\ \mathbf{R}_{21}(k) & \mathbf{R}_{22}(k) & 0 \\ \mathbf{R}_{31}(k) & \mathbf{R}_{32}(k) & \mathbf{R}_{33}(k) \end{bmatrix}^T. \tag{15}$$

Therefore, doing the singular value decomposition on $\mathcal{Q}$ yields the estimate of $[-\mathcal{O}_f^{\perp}\mathcal{H}_f \quad \mathcal{O}_f^{\perp}]$, as described

$$\begin{aligned} \mathcal{Q} &= \begin{bmatrix} \mathfrak{U} & \mathfrak{U}^{\perp} \end{bmatrix} \begin{bmatrix} \mathfrak{F} & \\ & \mathfrak{F}^{\perp} \end{bmatrix} \begin{bmatrix} \mathfrak{V}^T \\ \mathfrak{V}^{\perp T} \end{bmatrix} \\ \widehat{\mathcal{O}_f^{\perp}\mathcal{H}_f} &= -\mathfrak{U}_{lf} \in \mathbb{R}^{(mf-n)\times lf} \\ \widehat{\mathcal{O}_f^{\perp}} &= \mathfrak{U}_{mf} \in \mathbb{R}^{(mf-n)\times mf} \end{aligned} \tag{16}$$

where $\mathfrak{U}$ is composed of the first $(lf+n)$ left singular vectors that correspond to the $(lf+n)$ largest singular values collected in $\mathfrak{F}$, $\mathfrak{U}^{\perp}$ consists of the remaining $(mf-n)$ left singular vectors, $\mathfrak{V}$ and $\mathfrak{V}^{\perp}$ contain the right singular vectors, and $\mathfrak{U}_{lf}$ and $\mathfrak{U}_{mf}$ represent the first lf columns and the last mf columns of $\mathfrak{U}^{\perp T}$, respectively.

Lemma 1: If the QR decomposition of $[\mathbf{S}_{k-1,f,\alpha-1}^T, \mathbf{U}_{k-f-2,p,\alpha-1}^T, \mathbf{Y}_{k-f-1,p,\alpha-1}^T, \mathbf{U}_{k-2,f,\alpha-1}^T, \mathbf{Y}_{k-1,f,\alpha-1}^T]^T$ at time instant $k-1$ is known, and $\check{\mathbf{y}}_f(k)$ and $\mathbf{R}_{ij}(k)$ $(i = 1, 2, \ldots, 5$ and $j = 1, 2, \ldots, 4)$ are delivered by (14), then

$$\mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k) = \mu\mathbf{R}_{55}(k-1)\mathbf{R}_{55}^T(k-1) + \check{\mathbf{y}}_f(k)\check{\mathbf{y}}_f^T(k) \tag{17}$$

$$\begin{aligned} &\left(\boldsymbol{\Upsilon}_f\mathbf{F}_{k,f,\alpha} + \mathcal{G}_f\mathbf{W}_{k,f,\alpha}\right)\left(\boldsymbol{\Upsilon}_f\mathbf{F}_{k,f,\alpha} + \mathcal{G}_f\mathbf{W}_{k,f,\alpha}\right)^T \\ &\quad = \mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k). \end{aligned} \tag{18}$$

Proof: For the sake of convenience, $\boldsymbol{\Xi}$ and $\boldsymbol{\Theta}$ are, respectively, defined as

$$\begin{aligned} \boldsymbol{\Xi} = \big[&\sqrt{\mu}\mathbf{R}_{51}(k-1) \quad \sqrt{\mu}\mathbf{R}_{52}(k-1) \quad \sqrt{\mu}\mathbf{R}_{53}(k-1) \\ &\sqrt{\mu}\mathbf{R}_{54}(k-1) \quad \sqrt{\mu}\mathbf{R}_{55}(k-1) \quad \mathbf{y}_f(k) \big] \end{aligned} \tag{19}$$

and

$$\begin{aligned} \boldsymbol{\Theta} = \big[&\mathbf{R}_{51}(k) \quad \mathbf{R}_{52}(k) \quad \mathbf{R}_{53}(k) \\ &\mathbf{R}_{54}(k) \quad \sqrt{\mu}\mathbf{R}_{55}(k-1) \quad \check{\mathbf{y}}_f(k) \big]. \end{aligned} \tag{20}$$

As a result, in light of (14), the following equation holds:

$$\boldsymbol{\Xi}\boldsymbol{\Phi}(k) = \boldsymbol{\Theta}. \tag{21}$$

Due to the fact that $\boldsymbol{\Phi}(k)$ is an orthogonal matrix, it can be seen from (21) that

$$\boldsymbol{\Xi}\boldsymbol{\Xi}^T = \boldsymbol{\Theta}\boldsymbol{\Theta}^T. \tag{22}$$

Besides, from (10), (19), and the definition of $\mathbf{Y}_{k,f,\alpha}$, it is known that

$$\begin{aligned} \mathbf{Y}_{k,f,\alpha}\mathbf{Y}_{k,f,\alpha}^T &= \begin{bmatrix} \sqrt{\mu}\mathbf{Y}_{k-1,f,\alpha-1} & \mathbf{y}_f(k) \end{bmatrix} \begin{bmatrix} \sqrt{\mu}\mathbf{Y}_{k-1,f,\alpha-1}^T \\ \mathbf{y}_f^T(k) \end{bmatrix} \\ &= \boldsymbol{\Xi}\boldsymbol{\Xi}^T. \end{aligned} \tag{23}$$

Then, substituting (22) into (23) yields

$$\mathbf{Y}_{k,f,\alpha}\mathbf{Y}_{k,f,\alpha}^T = \boldsymbol{\Theta}\boldsymbol{\Theta}^T. \tag{24}$$

Moreover, by (10), $\mathbf{Y}_{k,f,\alpha}\mathbf{Y}_{k,f,\alpha}^T$ is also written as

$$\begin{aligned} \mathbf{Y}_{k,f,\alpha}\mathbf{Y}_{k,f,\alpha}^T = &\mathbf{R}_{51}(k)\mathbf{R}_{51}^T(k) + \mathbf{R}_{52}(k)\mathbf{R}_{52}^T(k) \\ &+ \mathbf{R}_{53}(k)\mathbf{R}_{53}^T(k) + \mathbf{R}_{54}(k)\mathbf{R}_{54}^T(k) \\ &+ \mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k). \end{aligned} \tag{25}$$

After that, from (20), (24), and (25), it is observed that (17) holds. Finally, on the basis of (11), (18) is gotten. So, this proof is completed. ■

Next, according to the above-obtained results, how to acquire the estimate of $\mathcal{O}_f^{\perp}\mathscr{H}_f$ will be discussed.

To this end, $\widehat{\mathcal{O}_f} \in \mathbb{R}^{mf\times n}$ is first obtained in terms of $\widehat{\mathcal{O}_f^{\perp}}$. As a result, by using the matrix representation in the MATLAB, the estimate of $\mathcal{O}_f^{\perp}\mathcal{H}_f(:, 1{:}\ l)$ is rewritten in the underlying way

$$\widehat{\mathcal{O}_f^{\perp}\mathcal{H}_f}(:, 1:l) = \widehat{\mathcal{O}_f^{\perp}} \begin{bmatrix} 0_{m\times n} \\ \widehat{\mathcal{O}_f}(1:m(f-1), 1:n) \end{bmatrix} \widehat{\mathbf{B_{d1}}}. \tag{26}$$

Then, by selecting a proper $f > n$ such that $mf \geq 2n$, $\widehat{\mathbf{B_{d1}}}$ can be estimated by

$$\widehat{\mathbf{B_{d1}}} = \left(\widehat{\mathcal{O}_f^{\perp}} \begin{bmatrix} 0_{m\times n} \\ \widehat{\mathcal{O}_f}(1:m(f-1), 1:n) \end{bmatrix} \right)^{\dagger} \times \widehat{\mathcal{O}_f^{\perp}\mathcal{H}_f}(:, 1:l) \tag{27}$$

which further leads to

$$\widehat{\mathcal{H}}_f(m+1:mf, 1:l) = \widehat{\mathcal{O}_f}(1:m(f-1), 1:n)\widehat{\mathbf{B_{d1}}} \tag{28}$$

where $\dagger$ represents the pseudo-inverse symbol. By letting $\widehat{\mathcal{H}}_f(m+1:mf, 1:l)$ be expressed in the form

$$\widehat{\mathcal{H}}_f(m+1:mf, 1:l) = \begin{bmatrix} \boldsymbol{\eta}_0^T & \boldsymbol{\eta}_1^T & \cdots & \boldsymbol{\eta}_{f-2}^T \end{bmatrix}^T \tag{29}$$

with $\boldsymbol{\eta}_i \in \mathbb{R}^{m\times l}$ $(i = 0, 1, \ldots, f-2)$, $\boldsymbol{\mathcal{M}}$ is defined as

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} [\boldsymbol{\eta}_0, 0_{m\times l}] & 0_{m\times 2l} & \cdots & 0_{m\times 2l} \\ [\boldsymbol{\eta}_1, -\boldsymbol{\eta}_0] & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{m\times 2l} \\ [\boldsymbol{\eta}_{f-2}, -\boldsymbol{\eta}_{f-3}] & \cdots & [\boldsymbol{\eta}_1, -\boldsymbol{\eta}_0] & [\boldsymbol{\eta}_0, 0_{m\times l}] \end{bmatrix}. \tag{30}$$

Furthermore, $\widehat{\mathcal{O}_f^{\perp}\mathcal{H}_f}(:, l+1 : fl)$ is denoted by

$$\widehat{\mathcal{O}_f^{\perp}\mathcal{H}_f}(:, l+1 : fl) = \begin{bmatrix} \boldsymbol{\vartheta}_1 & \boldsymbol{\vartheta}_2 & \cdots & \boldsymbol{\vartheta}_{f-1} \end{bmatrix} \tag{31}$$

so that $\boldsymbol{\mathcal{N}}$ is defined as

$$\boldsymbol{\mathcal{N}} = \begin{bmatrix} [0_{(mf-n)\times l}, \boldsymbol{\vartheta}_1] & [0_{(mf-n)\times l}, \boldsymbol{\vartheta}_2] & \cdots \\ [0_{(mf-n)\times l}, \boldsymbol{\vartheta}_{f-1}] & 0_{(mf-n)\times 2l} \end{bmatrix} \tag{32}$$

where $\boldsymbol{\vartheta}_j \in \mathbb{R}^{(mf-n)\times l} (j = 1, 2, \ldots, f-1)$. As a result

$$\widehat{\mathcal{O}_f^{\perp}\mathscr{H}_f} = \boldsymbol{\mathcal{N}} + \widehat{\mathcal{O}_f^{\perp}} \begin{bmatrix} 0_{m\times 2l(f-1)} & 0_{m\times 2l} \\ \boldsymbol{\mathcal{M}} & 0_{m(f-1)\times 2l} \end{bmatrix}. \tag{33}$$

On the basis of the above-obtained results, a data-driven recursive identification algorithm for the construction of the attack detector is summarized below.

Remark 3: By formulating an extended input vector $\bar{\mathbf{u}}[k]$ (that contains $\mathbf{u}[k]$ and $\mathbf{u}[k-1]$) and then rewriting the dynamics (4) in terms of this extended input vector, a system without delay can be obtained. Even though the system is exploited, the least squares estimates of $\mathcal{O}_f^{\perp}$ and $\mathcal{O}_f^{\perp}\mathscr{H}_f$ still cannot be gotten via the method developed in [19]. This is because the Hankel matrix $\bar{\mathbf{U}}_{k,f,\alpha}$ consisting of $\bar{\mathbf{u}}[k]$ is not of full row rank even under an input excitation condition. In contrast, $\mathbf{U}_{k-1,f,\alpha}$ in (7) is a full-row-rank matrix, so that, under the noise-corrupted case, the consistent estimation of $\mathcal{O}_f^{\perp}$ and $\mathcal{O}_f^{\perp}\mathscr{H}_f$ is achieved with Algorithm 1.

Here, suppose that the covariances of both $\mathbf{w}[k]$ and $\mathbf{f}[k]$ are stationary. Thus, from Lemma 1, it follows that:

$$\begin{aligned} &\lim_{k\to\infty} \mathcal{E}\big(\mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k)\big) \\ &\quad = \lim_{k\to\infty} \mathcal{E}\Big(\big(\boldsymbol{\Upsilon}_f\mathbf{F}_{k,f,\alpha} + \boldsymbol{\mathcal{G}}_f\mathbf{W}_{k,f,\alpha}\big)\big(\boldsymbol{\Upsilon}_f\mathbf{F}_{k,f,\alpha} + \boldsymbol{\mathcal{G}}_f\mathbf{W}_{k,f,\alpha}\big)^T\Big) \\ &\quad = \frac{1}{1-\mu}\Big(\boldsymbol{\mathcal{G}}_f\mathcal{E}\Big(\mathbf{w}_f(k)\mathbf{w}_f^T(k)\Big)\boldsymbol{\mathcal{G}}_f^T + \boldsymbol{\Upsilon}_f\mathcal{E}\Big(\mathbf{f}_f(k)\mathbf{f}_f^T(k)\Big)\boldsymbol{\Upsilon}_f^T\Big) \end{aligned}$$

where $\mathcal{E}(\cdot)$ is regarded as the statistical expectation operator. Consequently, M_R in Step 4 of Algorithm 1 is used for terminating this algorithm.

Besides, it follows from (5), (34), (35), (36), and (37) that

$$\boldsymbol{\epsilon}[k+1] = \mathbf{A_o}\boldsymbol{\epsilon}[k] + \mathbf{T_n}\mathbf{v}[k] \tag{38}$$

$$r[k] = \mathbf{C_o}\boldsymbol{\epsilon}[k] \tag{39}$$

where $\boldsymbol{\epsilon}[k] = \check{\mathbf{x}}[k] - \boldsymbol{\chi}[k]$. Equations (38) and (39) indicate that, provided $\mathbf{w}[k] \equiv 0$ and $\mathbf{f}[k] \equiv 0$, $\boldsymbol{\chi}[k]$ must converge to $\check{\mathbf{x}}[k]$ through n time steps since all the eigenvalues of $\mathbf{A_o}$ are zero.

So far, we have achieved the data-driven design of the attack detector. Our remaining task is to devise the speed servo controller so as to eventually build up the benchmark platform for testing stealthy attack technology. To this end, the tracking control problem considering performance optimization is to be addressed in the underlying section.

Algorithm 1 Design of the Attack Detector via the Closed-Loop Data in CPS

S1: Select a proper $0 \ll \mu < 1$. Then, let $\mathbf{R}_{ii}(k-1)$ $(i = 1, 2, \ldots, 5)$ in the LHS of (14) be the identity matrix, and the other matrix blocks be filled with random numbers $\in (0, 1)$.

S2: Gather the new data, including $\mathbf{s}_f(k)$, $\mathbf{u}_p(k-f-1)$, $\mathbf{y}_p(k-f)$, $\mathbf{u}_f(k-1)$ and $\mathbf{y}_f(k)$.

S3: Referring to Appendix B in [26], the sequence of Givens rotations $\boldsymbol{\Phi}(k)$ is computed. Then, $\mathbf{R}_{ij}(k)$ and $\check{\mathbf{y}}_f(k)$ are obtained by (14), where $i = 1, 2, \ldots, 5$ and $j = 1, 2, \ldots, 4$.

S4: Taking advantage of (17), $\mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k)$ is acquired. If $|\text{trace}(\mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k)) - M_R|/M_R < \xi$, go to S5, otherwise let $k = k+1$ and go to S2, where ξ is the convergence precision prescribed in advance, and M_R is the mean value of $\text{trace}(\mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k))$ obtained with the large enough k.

S5: $\widehat{\mathcal{O}_f^{\perp}}$ and $\widehat{\mathcal{O}_f^{\perp}\mathcal{H}_f}$ are derived via (15) and (16).

S6: By (27) and (28), $\widehat{\mathcal{H}}_f(m+1 : mf, 1 : l)$ is attained.

S7: According to (30) and (32), $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{N}}$ are built up.

S8: $\widehat{\mathcal{O}_f^{\perp}\mathscr{H}_f}$ is gotten in terms of (33).

S9: With the aid of $\widehat{\mathcal{O}_f^{\perp}}$ and $\widehat{\mathcal{O}_f^{\perp}\mathscr{H}_f}$, the observability canonical form of (4)-(5) is achieved as

$$\begin{aligned} \check{\mathbf{x}}[k+1] &= \mathbf{A_M}\check{\mathbf{x}}[k] + \mathbf{B_{o1}}\mathbf{u}[k-1] + \mathbf{B_{o2}}\mathbf{u}[k] \\ &\quad + \mathbf{T_n}\mathbf{v}[k], \end{aligned} \tag{34}$$

$$\mathbf{y}[k] = \boldsymbol{\Pi}^{-1}\mathbf{C_M}\check{\mathbf{x}}[k] \tag{35}$$

and the attack detector is constructed as

$$\begin{aligned} \boldsymbol{\chi}[k+1] &= \mathbf{A_o}\boldsymbol{\chi}[k] + \mathbf{B_{o1}}\mathbf{u}[k-1] + \mathbf{B_{o2}}\mathbf{u}[k] \\ &\quad + \mathbf{L_o}\boldsymbol{\pi}\mathbf{y}[k], \end{aligned} \tag{36}$$

$$r[k] = \boldsymbol{\pi}\mathbf{y}[k] - \mathbf{C_o}\boldsymbol{\chi}[k] \tag{37}$$

where $\check{\mathbf{x}}[k] = \mathbf{T_n}\mathcal{F}\mathbf{x}[k]$, $\mathcal{O}_f = \widehat{\mathcal{O}_f}\mathcal{F}$, $\mathcal{F}$ is a full-rank state transition matrix, $\mathbf{v}[k] = \mathcal{F}\mathbf{E_d}\mathbf{f}[k] + \mathcal{F}\mathbf{K_d}\mathbf{w}[k]$, and the realizations of $\mathbf{A_M}$, $\mathbf{C_M}$, $\mathbf{A_o}$, $\mathbf{B_{o1}}$, $\mathbf{B_{o2}}$, $\mathbf{L_o}$, $\mathbf{C_o}$, $\mathbf{T_n}$, $\boldsymbol{\Pi}$, $\boldsymbol{\pi}$, and the threshold value J_{th} refer to [19].

S10: It is believed that, if $J_k > J_{\text{th}}$, then an attack is detected, where $J_k = \sigma_r^{-1}r^2[k]$ and σ_r is the variance of $r[k]$.

B. Tracking Control Design for Performance Optimization

Take account of the performance index of the form

$$\mathcal{J}_k = \sum_{j=1}^{k}\big(\mathbf{u}^T[j-1]\mathbf{Q_u}\mathbf{u}[j-1] + \boldsymbol{\eta}^T[j]\mathbf{Q}_{\boldsymbol{\eta}}\boldsymbol{\eta}[j] + \check{\mathbf{x}}^T[j]\mathbf{Q_x}\check{\mathbf{x}}[j]\big) \tag{40}$$

where $\mathbf{Q_u} > 0$, $\mathbf{Q}_{\boldsymbol{\eta}} \geq 0$, and $\mathbf{Q_x} \geq 0$ are the weighting matrices given beforehand, $\boldsymbol{\eta}[k] = \sum_{j=0}^{k-1}\boldsymbol{\epsilon}[j]$, and

$\boldsymbol{\varepsilon}[k] = \mathbf{s}[k] - \mathbf{y}[k]$. By (34), the following augmented system is first constructed:

$$\mathbf{z}[k+1] = \breve{\mathbf{A}}\mathbf{z}[k] + \breve{\mathbf{B}}\mathbf{u}[k] + \mathscr{E}\mathbf{d}[k] \tag{41}$$

with

$$\mathbf{z}[k] = \begin{bmatrix} \mathbf{u}^T[k-1] & \boldsymbol{\eta}^T[k] & \breve{\mathbf{x}}^T[k] \end{bmatrix}^T,\ \mathbf{d}[k] = \begin{bmatrix} \mathbf{v}[k] \\ \mathbf{s}[k] \end{bmatrix}$$

$$\breve{\mathbf{A}} = \begin{bmatrix} 0_{l\times l} & 0_{l\times m} & 0_{l\times n} \\ 0_{m\times l} & \mathbf{I}_m & -\boldsymbol{\Pi}^{-1}\mathbf{C_M} \\ \mathbf{B_{o1}} & 0_{n\times m} & \mathbf{A_M} \end{bmatrix}, \breve{\mathbf{B}} = \begin{bmatrix} \mathbf{I}_l \\ 0_{m\times l} \\ \mathbf{B_{o2}} \end{bmatrix}$$

and $\mathscr{E} = \begin{bmatrix} 0_{l\times n} & 0_{l\times m} \\ 0_{m\times n} & \mathbf{I}_m \\ \mathbf{T_n} & 0_{n\times m} \end{bmatrix}$.

Then, substituting the control policy

$$\mathbf{u}[k] = \mathbf{L_u}\mathbf{u}[k-1] + \mathbf{L}_{\boldsymbol{\eta}}\boldsymbol{\eta}[k] + \mathbf{L}_{\boldsymbol{\chi}}\boldsymbol{\chi}[k] \tag{42}$$

into (41) with $\mathbf{w}[k] = 0$ and $\mathbf{f}[k] = 0$ yields the closed-loop augmented system of the form

$$\mathbf{z}[k+1] = \left(\breve{\mathbf{A}} + \breve{\mathbf{B}}\breve{\mathbf{L}}\right)\mathbf{z}[k] + \breve{\mathbf{E}}\mathbf{s}[k] \tag{43}$$

where $\breve{\mathbf{L}} = \begin{bmatrix} \mathbf{L_u} & \mathbf{L}_{\boldsymbol{\eta}} & \mathbf{L}_{\boldsymbol{\chi}} \end{bmatrix}$ and

$$\breve{\mathbf{E}} = \begin{bmatrix} 0_{l\times m} \\ \mathbf{I}_m \\ 0_{n\times m} \end{bmatrix}.$$

Furthermore, the performance output is defined as

$$\boldsymbol{\zeta}[k] = \breve{\mathbf{C}}\mathbf{z}[k] \tag{44}$$

where

$$\breve{\mathbf{C}} = \begin{bmatrix} \mathbf{Q_u}^{1/2} & 0_{l\times m} & 0_{l\times n} \\ 0_{m\times l} & \mathbf{Q}_{\boldsymbol{\eta}}^{1/2} & 0_{m\times n} \\ 0_{n\times l} & 0_{n\times m} & \mathbf{Q_x}^{1/2} \end{bmatrix}.$$

As a result, the following theorem is given to find $u[k]$ guaranteeing the tracking performance.

Theorem 1: For $\gamma > 0$ prescribed in advance, it is supposed that there exists the symmetric $\boldsymbol{\Lambda} \in \mathbb{R}^{\kappa\times\kappa}$ and $\mathbf{G} \in \mathbb{R}^{\kappa\times\kappa}$, together with the asymmetric $\mathbf{V} \in \mathbb{R}^{l\times\kappa}$ so that the underlying linear matrix inequalities hold

$$\begin{bmatrix} -\mathbf{G} & 0_{\kappa\times m} & \mathbf{G}\breve{\mathbf{A}}^T + \mathbf{V}^T\breve{\mathbf{B}}^T & \mathbf{G}\breve{\mathbf{C}}^T \\ * & -\gamma\mathbf{I}_m & \breve{\mathbf{E}}^T & 0_{m\times\kappa} \\ * & * & -\mathbf{G} & 0_{\kappa\times\kappa} \\ * & * & * & -\mathbf{I}_\kappa \end{bmatrix} < 0 \tag{45}$$

$$\begin{bmatrix} \mathbf{G} & \mathbf{I}_\kappa \\ * & \boldsymbol{\Lambda} \end{bmatrix} > 0 \tag{46}$$

where $\kappa = l + m + n$. Then, the control law (42) with $\breve{\mathbf{L}} = \mathbf{V}\mathbf{G}^{-1}$ stabilizes the closed-loop augmented system (43). Moreover, an upper bound of (40) has the form

$$\mathcal{J}_k \le \mathbf{z}^T[1]\boldsymbol{\Lambda}\mathbf{z}[1] + \gamma\sum_{j=1}^{k}\mathbf{s}^T[j]\mathbf{s}[j] \tag{47}$$

where γ corresponds to the H-infinity norm of the transfer function from $\mathbf{s}[k]$ to $\boldsymbol{\zeta}[k]$. Besides, this upper bound can be minimized by tackling the underlying optimization problem via the LMI toolbox:

min trace($\boldsymbol{\Lambda}$), subject to (45) and (46).

Proof: Employing some classical results in LMI theory [32], the theorem can be derived. ■

C. Software Design for the Benchmark Platform

By means of the above-obtained results, we have developed a software program (i.e., Program 3 shown in Fig. 3) for both the attack detector and the speed servo controller.

The software environment of the benchmark platform for testing stealthy attack approaches has also been displayed in Fig. 3, where five programs have been designed and the media access control (MAC) and IP addresses of each computer presented. Programs 1 and 2 run on the client computer. Program 1 is devised with the MATLAB/Simulink real-time windows target (RTWT) software, and utilized for accessing the PCI-1711U so as to get the speed information $\tilde{y}$ of the motor and apply the control command $\tilde{u}$ (coming from the server computer) to the dc machine. Besides, it is noted that the TCP is not directly supported by the RTWT software executed under the Simulink external mode, while the UDP is. Thus, through resorting to MATLAB scripts, Program 2 is devised, which actually constructs the interface between the UDP and TCP ports. As a result, first, Program 1 transmits $\tilde{y}$ to Program 2 by means of the UDP ports 13002 and 59998. Then, by making use of the local TCP port 48695 and the remote TCP port 8080, Program 2 sends $\tilde{y}$ to Program 3, which operates on the server computer and is developed via MATLAB scripts. After that, based on $\tilde{y}$, $\tilde{u}$ is calculated by Program 3 and then dispatched to Program 2, employing the TCP ports 8080 and 48695. Subsequently, Program 2 gives $\tilde{u}$ to Program 1 with the aid of the UDP ports 59997 and 12001. Finally, Program 1 implements the received $\tilde{u}$ on the dc motor, driving the 12-bit digital-to-analog converter of PCI-1711U by virtue of the RTWT software. Note that, although the data exchange between Programs 1 and 2 is achieved via the UDP, the exchange is deemed to be reliable in this paper because it occurs on the same computer (i.e., client computer), that is, the exchanged information does not go through the network adapter and any external network.

III. Stealthy Attack on TCP/IP-Based CPS With Unknown Dynamics

In the following section, by discovering and then using a potential security flaw in TCP/IP-based CPS, we will present a methodology for eavesdropping, falsification, and blocking of information so as to remove Assumptions 2 and 3 mentioned in Section I.

A. Eavesdropping, Falsification, and Blocking of Information in TCP/IP-Based CPS

Since the Ethernet hub operates by repeating the packets received from one of its ports to all other ports, the network interface controller (NIC) of the attacker's computer depicted in Fig. 2 is capable of receiving the frames transmitted between the client and server computers. However, under the widely

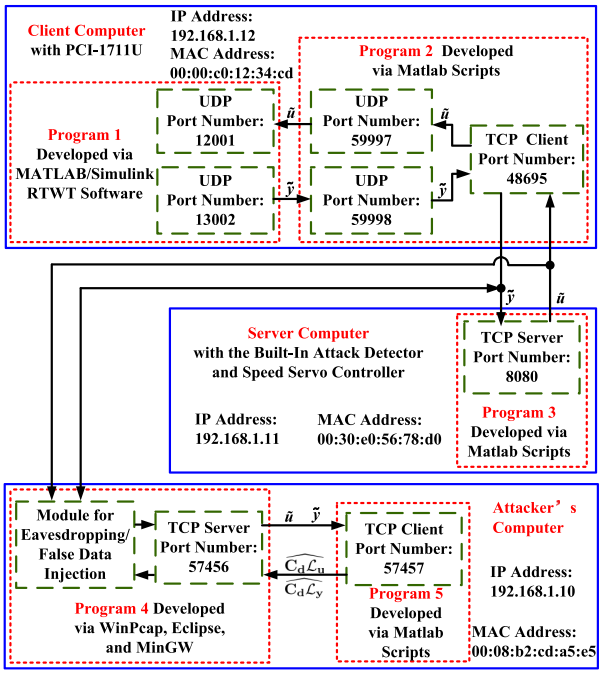

Fig. 3. Software environment of the benchmark platform for testing stealthy attack technology.

used nonpromiscuous mode, the NIC normally drops these frames because the destination MAC address included in each frame is not the MAC address of the NIC. For this reason, through use of the windows packet capture library WinPcap, the NIC of the attacker's computer is configured for the promiscuous mode, so that the NIC passes all traffic that it receives to the central processing unit (CPU) rather than passing only the frames related to its MAC address. Furthermore, without the interposition of the protocol processing executed by the operating system, Program 4 shown in Fig. 3 can directly eavesdrop raw packets (i.e., $\tilde{y}$ and $\tilde{u}$) on the network via WinPcap. It is well-known that, by utilizing the sequence and acknowledgment numbers, the TCP/IP achieves the error control at the transport layer to ensure that the packets are reliably exchanged between the TCP client and server. If two received packets have the same sequence number, the receiving transport layer deems that a duplicate packet arises, which subsequently can be automatically discarded by the receiver. Considering the aforementioned error control mechanism, the underlying strategy is designed to block the TCP/IP-based communication and send the falsified data.

In addition to intercepting $\tilde{y}$ and $\tilde{u}$ with WinPcap, the port numbers and the MAC and IP addresses of the client and server computers are ascertained by means of the aforesaid information eavesdropping techniques. Moreover, Program 4 extracts the TCP data length and the sequence and acknowledgment numbers (denoted by ℓ_{td}, $\aleph_{sq}$, and $\aleph_{ac}$, respectively) from the data packet sent by the server computer at the instant before the attack is launched. After that and before the TCP client supplies new sensor data to the server, the stealthy attack is started, that is, Program 4 transmits a forged packet (which mainly consists of the port number and the MAC and IP addresses of the client computer, the header checksum of the IP datagram, the sequence number $= \aleph_{ac}$, the acknowledgment number $= \aleph_{sq} + \ell_{td}$, the false sensor information $\ddot{y}[k]$, and the checksum of the TCP segment) to the server computer with the aid of WinPcap. Subsequently, Program 4 intercepts the control command included in the packet transmitted by the TCP server so as to calculate $\ddot{y}[k+1]$. Note that, after launching the attack, a forged packet is given to the server every T s, which is the sampling time of the networked control system. In summary, a false packet is provided to the server by the attacker's computer before a packet including an actual measurement is sent by the client computer. As a result, since these two packets have the same sequence number, the packet coming from the client is a duplicate packet for the server and therefore is silently discarded. Consequently, the data transmission from the TCP client to the TCP server is obstructed.

Every time when a duplicate packet arrives, besides discarding it, the server immediately transmits an acknowledgment (ACK) pointing out the next expected packet. This leads to a traffic surge. Hence, the phenomenon appears that the packet from the server to the client is missing. Nevertheless, the error is not corrected through retransmission due to the following reason: during implementing the stealthy attack, Program 4 immediately sends an ACK for each packet delivered by the TCP server, so that the packet corresponding to the ACK is thought of as having arrived soundly and safely by the server. Therefore, the packet copy located in the buffer of the server is purged. For this reason, the retransmission cannot be carried out. As a result, after the above error caused by the stealthy attack arises, the data packets that are delivered by the server and reach the client computer are out of order. Thus, they are stored temporarily and flagged as out-of-order packets. However, the TCP ensures that no out-of-order information is given to the process working at the application layer. In other words, the stealthy attack also blocks the traffic from the server to the client.

As depicted in Fig. 3, the stealthy attack software running on attacker's computer is composed of Programs 4 and 5. Program 4 is designed with the WinPcap, the development environment Eclipse, and the MinGW compiler. Program 4 is in charge of eavesdropping the communication data between the client and server computers, and is responsible for forging the MAC address, IP address, and port number of the client computer so as to transmit the falsified data to the server computer.

In addition, due to the fact that it is difficult to perform matrix operations under the Eclipse, Program 5 is designed based on MATLAB scripts, which are used for identifying the dynamic characteristic matrix of TCP/IP-based CPS from the eavesdropped data $\tilde{y}$ and $\tilde{u}$. These data need to come from Program 4. Furthermore, the dynamic characteristic matrix delivered by Program 5 needs to be passed to Program 4, which simulates (via the obtained dynamic characteristic matrix) sensor outputs under the control inputs calculated by the server computer, and then sends the simulated outputs back to the server computer such that the attack goes undetected by the attack detector. To this end, Program 4 constructs a TCP server (with the port number 57456) via the Windows sockets library. The corresponding TCP client (with the port number

57457) is built up by Program 5. As a result, from Fig. 3, it can be observed that $\tilde{y}, \tilde{u}$, and the dynamic characteristic matrix $[\widehat{\mathbf{C_d}\mathcal{L}_\mathbf{u}} \quad \widehat{\mathbf{C_d}\mathcal{L}_\mathbf{y}}]$ are transmitted between Programs 4 and 5.

To remove Assumption 1 in Section I, the next section will give a method for the recursive identification of the above dynamic characteristic matrix via the eavesdropped closed-loop data.

B. Identification of the Dynamic Characteristic Matrix via the Eavesdropped Information

To this end, the plant under consideration is first modeled as the so-called innovation form

$$\hat{\mathbf{x}}[k+1] = \mathbf{A_d}\hat{\mathbf{x}}[k] + \mathbf{B_{d1}}\mathbf{u}[k-1] + \mathbf{B_{d2}}\mathbf{u}[k] + \mathcal{K}(\mathbf{e}[k] + \mathbf{h}[k]) \tag{48}$$

$$\mathbf{y}[k] = \mathbf{C_d}\hat{\mathbf{x}}[k] + \mathbf{e}[k] \tag{49}$$

$$\hat{\mathbf{y}}[k] = \mathbf{y}[k] + \mathbf{h}[k] \tag{50}$$

where $\mathbf{e}[k]$ is the innovation, $\mathbf{h}[k]$ is an added zero-mean pseudo-random number, and $\mathcal{K}$ is the Kalman filter gain corresponding to $\mathbf{e}[k] + \mathbf{h}[k]$.

From (48)–(50), it follows that:

$$\hat{\mathbf{x}}[k] = \mathfrak{A}_\mathbf{K}^p\hat{\mathbf{x}}[k-p] + \begin{bmatrix}\mathcal{L}_\mathbf{u} & \mathcal{L}_\mathbf{y}\end{bmatrix}\begin{bmatrix}\mathbf{u}_{p+1}^T(k-1) & \hat{\mathbf{y}}_p^T(k-1)\end{bmatrix}^T \tag{51}$$

where

$$\mathcal{L}_\mathbf{u} = \begin{bmatrix}\mathfrak{A}_\mathbf{K}^{p-1}\mathbf{B_{d1}} & \mathfrak{A}_\mathbf{K}^{p-2}\mathfrak{B}_\mathbf{K} & \cdots & \mathfrak{A}_\mathbf{K}\mathfrak{B}_\mathbf{K} & \mathfrak{B}_\mathbf{K} & \mathbf{B_{d2}}\end{bmatrix}$$

$$\mathcal{L}_\mathbf{y} = \begin{bmatrix}\mathfrak{A}_\mathbf{K}^{p-1}\mathcal{K} & \mathfrak{A}_\mathbf{K}^{p-2}\mathcal{K} & \cdots & \mathfrak{A}_\mathbf{K}\mathcal{K} & \mathcal{K}\end{bmatrix}$$

$$\mathfrak{A}_\mathbf{K} = \mathbf{A_d} - \mathcal{K}\mathbf{C_d}$$

and $\mathfrak{B}_\mathbf{K} = \mathfrak{A}_\mathbf{K}\mathbf{B_{d2}} + \mathbf{B_{d1}}$.

Here, the eigenvalues of $\mathfrak{A}_\mathbf{K}$ are supposed to be strictly inside the unit circle. Under this assumption, $\mathfrak{A}_\mathbf{K}^p \to 0$ as $p \to \infty$. As a result, by selecting a sufficiently large p, and on the basis of (49)–(51), the underlying I/O model is built up

$$\hat{\mathbf{Y}}_{k,1,\beta} = \mathbf{C_d}\begin{bmatrix}\mathcal{L}_\mathbf{u} & \mathcal{L}_\mathbf{y}\end{bmatrix}\begin{bmatrix}\mathbf{U}_{k-1,p+1,\beta} \\ \hat{\mathbf{Y}}_{k-1,p,\beta}\end{bmatrix} + \mathbf{E}_{k,1,\beta} + \mathbf{H}_{k,1,\beta} \tag{52}$$

with $\beta = k - p - 1$. Along the same line as the one shown in (14) and (17), we can attain

$$\begin{bmatrix}\sqrt{\mu}\mathcal{R}_{11}(k-1) & 0 & 0 \\ \sqrt{\mu}\mathcal{R}_{21}(k-1) & \sqrt{\mu}\mathcal{R}_{22}(k-1) & 0 \\ \sqrt{\mu}\mathcal{R}_{31}(k-1) & \sqrt{\mu}\mathcal{R}_{32}(k-1) & \sqrt{\mu}\mathcal{R}_{33}(k-1) \\ \mathbf{u}_{p+1}(k-1) \\ \hat{\mathbf{y}}_p(k-1) \\ \hat{\mathbf{y}}[k]\end{bmatrix}$$

$$\mathbf{\Phi}(k) = \begin{bmatrix}\mathcal{R}_{11}(k) & 0 & 0 & 0 \\ \mathcal{R}_{21}(k) & \mathcal{R}_{22}(k) & 0 & 0 \\ \mathcal{R}_{31}(k) & \mathcal{R}_{32}(k) & \sqrt{\mu}\mathcal{R}_{33}(k-1) & \check{\mathbf{y}}[k]\end{bmatrix} \tag{53}$$

$$\text{and } \mathcal{R}_{33}(k)\mathcal{R}_{33}^T(k) = \mu\mathcal{R}_{33}(k-1)\mathcal{R}_{33}^T(k-1) + \check{\mathbf{y}}[k]\check{\mathbf{y}}^T[k] \tag{54}$$

Algorithm 2 (Iterative Identification of the Dynamic Characteristic Matrix via the Eavesdropped Information)

- **S1:** Let $\mathcal{R}_{ii}(k-1)$ ($i = 1, 2,$ and 3) in the LHS of (53) be the identity matrix, and the other matrix blocks be filled with random numbers $\in (0, 1)$.
- **S2:** $\hat{\mathbf{y}}[k]$ is gotten by injecting $\mathbf{h}[k]$ into $\mathbf{y}[k]$. Then, the eavesdropped data are arranged as $\mathbf{u}_{p+1}(k-1)$ and $\hat{\mathbf{y}}_p(k-1)$.
- **S3:** $\mathcal{R}_{11}(k)$, $\mathcal{R}_{21}(k)$, $\mathcal{R}_{22}(k)$, $\mathcal{R}_{31}(k)$, $\mathcal{R}_{32}(k)$, and $\check{\mathbf{y}}(k)$ are calculated by (53).
- **S4:** $\mathcal{R}_{33}(k)\mathcal{R}_{33}^T(k)$ is attained through resorting to (54). If $|\text{trace}(\mathcal{R}_{33}(k)\mathcal{R}_{33}^T(k)) - \bar{M}_R|/\bar{M}_R < \xi$, go to S5, otherwise let $k = k+1$ and go to S2, where ξ is the convergence precision prescribed in advance, and $\bar{M}_R$ is the mean value of $\text{trace}(\mathcal{R}_{33}(k)\mathcal{R}_{33}^T(k))$ obtained with the large enough k.
- **S5:** On the basis of (55), the dynamic characteristic matrix $[\widehat{\mathbf{C_d}\mathcal{L}_\mathbf{u}} \quad \widehat{\mathbf{C_d}\mathcal{L}_\mathbf{y}}]$ is derived.

where $\mathcal{R}_{ij}(k-1)$ ($i, j = 1, 2,$ or 3) is the QR decomposition of $[\mathbf{U}_{k-2,p+1,\beta-1}^T, \hat{\mathbf{Y}}_{k-2,p,\beta-1}^T, \hat{\mathbf{Y}}_{k-1,1,\beta-1}^T]^T$ at time instant $k-1$. Subsequently, based on (52) and (53), the estimate of the dynamic characteristic matrix $[\mathbf{C_d}\mathcal{L}_\mathbf{u} \quad \mathbf{C_d}\mathcal{L}_\mathbf{y}]$ is presented in the form

$$\begin{bmatrix}\widehat{\mathbf{C_d}\mathcal{L}_\mathbf{u}} & \widehat{\mathbf{C_d}\mathcal{L}_\mathbf{y}}\end{bmatrix} = \begin{bmatrix}\mathcal{R}_{31}^T(k) \\ \mathcal{R}_{32}^T(k)\end{bmatrix}^T\mathcal{R}_{11,22}^\dagger \tag{55}$$

where

$$\mathcal{R}_{11,22} = \begin{bmatrix}\mathcal{R}_{11}(k) & 0 \\ \mathcal{R}_{21}(k) & \mathcal{R}_{22}(k)\end{bmatrix}. \tag{56}$$

In summary, an approach to identifying recursively $[\mathbf{C_d}\mathcal{L}_\mathbf{u} \quad \mathbf{C_d}\mathcal{L}_\mathbf{y}]$ is given as follows.

It is worth pointing out that, due to the following reason, $\mathbf{h}[k]$ needs to be employed during iterating steps 2–4 of Algorithm 2.

By means of (48)–(50), an extended state space model is acquired as

$$\hat{\mathbf{Y}}_{k-1,p,\beta} = \mathcal{O}_p\hat{\mathbf{X}}_{k-p,1,\beta} + \mathcal{H}_p\mathbf{U}_{k-2,p,\beta} + \mathbf{\Xi}_p\big(\mathbf{E}_{k-1,p,\beta} + \mathbf{H}_{k-1,p,\beta}\big) \tag{57}$$

where $\mathcal{O}_p$ can be expressed by replacing f in the matrix $\mathcal{O}_f$ with p, and

$$\mathbf{\Xi}_p = \begin{bmatrix}\mathbf{I}_m & 0 & \cdots & 0 \\ \mathbf{C_d}\mathcal{K} & \mathbf{I}_m & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mathbf{C_d}\mathbf{A_d}^{p-2}\mathcal{K} & \cdots & \mathbf{C_d}\mathcal{K} & \mathbf{I}_m\end{bmatrix}.$$

From (57) without $\mathbf{h}[k]$, it is easily seen that

$$\begin{bmatrix}\mathbf{U}_{k-1,p+1,\beta} \\ \mathbf{Y}_{k-1,p,\beta}\end{bmatrix} = \begin{bmatrix}\mathbf{I}_{lp} & 0_{lp\times l} & 0_{lp\times n} \\ 0_{l\times lp} & \mathbf{I}_l & 0_{l\times n} \\ \mathcal{H}_p & 0_{mp\times l} & \mathcal{O}_p\end{bmatrix}\begin{bmatrix}\mathbf{U}_{k-2,p,\beta} \\ \mathbf{U}_{k-1,1,\beta} \\ \hat{\mathbf{X}}_{k-p,1,\beta}\end{bmatrix} + \begin{bmatrix}0_{lp\times mp} \\ 0_{l\times mp} \\ \mathbf{\Xi}_p\end{bmatrix}\mathbf{E}_{k-1,p,\beta}. \tag{58}$$

Since $\mathbf{y}[k]$ could have the relatively large value and is often processed with filters (e.g., a second-order low-pass digital filter with the cutoff frequency of 1 Hz in the following experiment), $\|\mathbf{e}[j]\| \ll \|\mathbf{y}[j]\|\ \forall j$. Therefore, the last term of (58) can be neglected. As a result, (58) shows that

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{U}_{k-1,p+1,\beta}\\ \mathbf{Y}_{k-1,p,\beta}\end{bmatrix}\right) = l(p+1)+n$$

that is, the matrix is not of full row rank although $\mathbf{u}[k]$ satisfies the so-called input excitation condition, so that $\boldsymbol{\mathcal{R}}_{11,22}$ does not have full row rank. Hence, $[\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{u}}\ \ \mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{y}}]$ cannot be uniquely estimated in terms of (55). To this end, as described in (50), $\mathbf{h}[k]$ needs to be injected into $\mathbf{y}[k]$.

With the aid of $[\widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{u}}}\ \ \widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{y}}}]$ delivered by Algorithm 2, $\acute{\mathbf{y}}[k]$ can be constructed as

$$\acute{\mathbf{y}}[k] = \begin{bmatrix}\widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{u}}} & \widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{y}}}\end{bmatrix}\begin{bmatrix}\mathbf{u}_{p+1}(k-1)\\ \mathbf{y}_p(k-1)\end{bmatrix} \tag{59}$$

which is used for deciding whether $\mathbf{y}[k]$ is predicted well and attaining the mean and variance of $\mathbf{y}[k]-\acute{\mathbf{y}}[k]$.

Then, the following $\ddot{\mathbf{y}}[k]$ is employed to simulate sensor outputs under the control inputs given by the server computer

$$\grave{\mathbf{y}}[k] = \begin{bmatrix}\widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{u}}} & \widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{y}}}\end{bmatrix}\begin{bmatrix}\mathbf{u}_{p+1}(k-1)\\ \grave{\mathbf{y}}_p(k-1)\end{bmatrix}$$
$$\ddot{\mathbf{y}}[k] = \grave{\mathbf{y}}[k] + \boldsymbol{\varpi}[k]$$

where $\boldsymbol{\varpi}[k]$ is produced randomly from a normal distribution with the above-obtained mean and variance of $\mathbf{y}[k]-\acute{\mathbf{y}}[k]$.

Finally, by sending $\ddot{\mathbf{y}}[k]$ back to the server computer via the methodology proposed in Section III-A, a malicious third party is able to interrupt the communication between the TCP server and client, and avoid triggering the alarm generated by the attack detector.

IV. Experimental Studies

To validate the effectiveness of the proposed data-driven stealthy attack method, an experiment is to be carried out on the benchmark platform built in Section II. Furthermore, the data-driven design procedures of both the attack detector and the tracking controller of the benchmark platform will be further clarified in the experiment.

The experimental parameters $\mu = 0.998$, $n = 2$, $l = 1$, $m = 1$, $f = 4$, $p = 40$, $Q_u = 11.56$, $Q_\eta = 1$, $Q_x = 0_{2\times 2}$, $\gamma = 3.46$, and $\xi = 4\%$ are employed here.

Before conducting this experiment, the PI controller $u[k] = K_P\varepsilon[k] + K_I\sum_{j=1}^{k}\varepsilon[j]$ with the gains $K_P = 10$ and $K_I = 1$ is assumed to have been designed by [33]. During the early stages of this experiment, the PI controller runs on the server computer; the pseudo-random numbers are used as the reference signal (i.e., the desired motor speed) to satisfy the input excitation condition required by Algorithm 1 [34]; and Algorithm 1 is executed to devise the attack detector via the closed-loop data. After Algorithm 1 quits, on the basis of Theorem 1 and the identification results delivered by Algorithm 1, the tracking control policy considering performance optimization is acquired and then is applied to the dc motor, instead of the aforementioned PI control law. The experimental results have been drawn in Figs. 4–7. From Fig. 4(a) and (b), it is observed that, under the aforesaid pseudo-random numbers, the desired velocity and the corresponding control command $\tilde{u}$ have large fluctuations. Moreover, $\tilde{u}$ is sent to the client computer through the network and recorded in Fig. 6(b). Under the action of $\tilde{u}$, the motor rotating speed fluctuates significantly, as can be seen from Fig. 6(a). The speed is also transmitted to the server computer and depicted in Fig. 4(a).

Given $(\sqrt{\mu})^{3000} = 0.0496$, the impact of the initial values of $\mathbf{R}_{ij}(k-1)$ in Algorithm 1 is deemed to have almost been eliminated after $k = 3000$ (i.e., the 600th second). Hence, $S_{um} = S_{um} + \operatorname{trace}(\mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k))$ is not calculated until $k > 3000$, where S_{um} is a scalar with an initial value of zero. When $k = 3500$ (namely, the 700th second), the mean value of $\operatorname{trace}(\mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k))$ is given by $M_R = S_{um}/500$. The evolution of $\operatorname{trace}(\mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k))$ and M_R is depicted in Fig. 5(a). At the 728.2th second, Algorithm 1 delivers $\widehat{\mathcal{O}_f^{\perp}}$ and $\widehat{\mathcal{O}_f^{\perp}\mathscr{H}_f}$ on account of $|\operatorname{trace}(\mathbf{R}_{55}(k)\mathbf{R}_{55}^T(k)) - M_R|/M_R < \xi$. Subsequently, the attack detector is built up as

$$\begin{aligned}\boldsymbol{\chi}[k+1] = {}& \begin{bmatrix}0 & 0\\ 1 & 0\end{bmatrix}\boldsymbol{\chi}[k] + \begin{bmatrix}-0.0142\\ 1.3352\end{bmatrix}u[k-1]\\ & + \begin{bmatrix}-1.3209\\ 0.0036\end{bmatrix}u[k] + \begin{bmatrix}-0.8474\\ 1.8140\end{bmatrix}y[k] \qquad (60)\\ r[k] = {}& y[k] - [0\ \ 1]\boldsymbol{\chi}[k] \qquad (61)\end{aligned}$$

and the speed servo control policy considering performance optimization is constructed from Theorem 1 as

$$\begin{aligned}u[k] = {}& -11.0128u[k-1] + 0.2889\eta[k]\\ & + \begin{bmatrix}-8.3339 & -8.5612\end{bmatrix}\boldsymbol{\chi}[k]. \qquad (62)\end{aligned}$$

Moreover, referring to [19, Algorithm 3], J_k and the threshold J_{th} are derived and displayed in Fig. 5(b). Thereafter, we believe that an attack arises provided that $J_k > J_{\text{th}}$. At the same time, the control law (62) and the reference signal of the form $s[k] = 1300 + 200\sin(2\pi k/1000)$ are applied to the motor system. As a result, by means of Fig. 4(a), it is demonstrated that the control policy delivered by Theorem 1 is capable of guaranteeing the tracking property.

At the 826.8th second, Programs 4 and 5 start working on the attacker's computer to eavesdrop the packets including the information about the control instruction and the motor velocity. Meanwhile, adding the normally distributed pseudo-random numbers with mean 0 and variance 1 to the obtained speed values, steps 1–4 of Algorithm 2 are executed to calculate $\boldsymbol{\mathcal{R}}_{ij}(k)$ $(i, j = 1, 2, \text{ or } 3)$. Here, the captured motor speeds are drawn in Fig. 7(a) and the convergence process of $\operatorname{trace}(\boldsymbol{\mathcal{R}}_{33}(k)\boldsymbol{\mathcal{R}}_{33}^T(k))$ depicted in Fig. 7(b), where $\operatorname{trace}(\boldsymbol{\mathcal{R}}_{33}(k)\boldsymbol{\mathcal{R}}_{33}^T(k))$ undergoes the similar procedure to the above-described one used for terminating Algorithm 1. That is to say, Step 5 of Algorithm 2 is not performed until the 1527th second, when $|\operatorname{trace}(\mathcal{R}_{33}(k)\mathcal{R}_{33}^T(k)) - \bar{M}_R|/\bar{M}_R < \xi$. Then, based on (55), the dynamic characteristic matrix $[\widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{u}}}\ \ \widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{y}}}]$ is attained. Within the ensuing 200 s, the predicted value $\acute{y}[k]$ of the actual motor speed is constructed with the aid of $[\widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{u}}}\ \ \widehat{\mathbf{C_d}\boldsymbol{\mathcal{L}}_{\mathbf{y}}}]$ and the past input–output data of the motor system eavesdropped over the network. Moreover, the

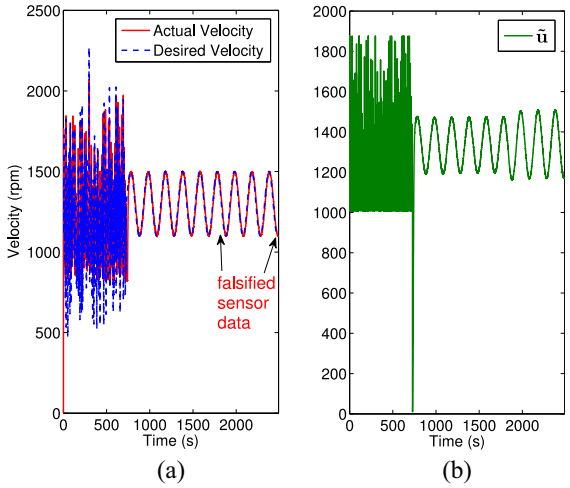

Fig. 4. Data recorded by the server computer, including the (a) actual and desired velocities of the motor and (b) control command.

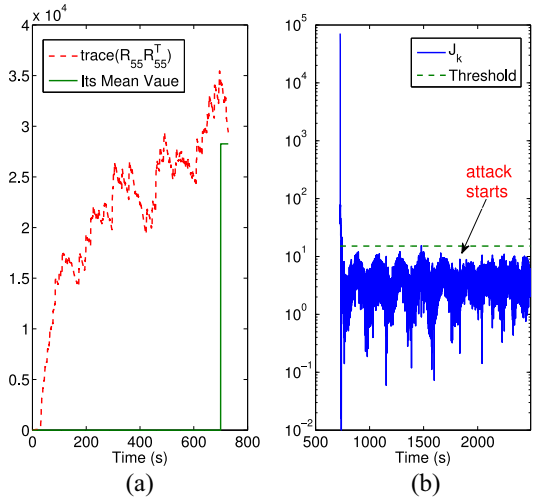

Fig. 5. Data recorded by the server computer, including the (a) trace of $R_{55}R_{55}^T$ and its mean value and (b) attack detection results.

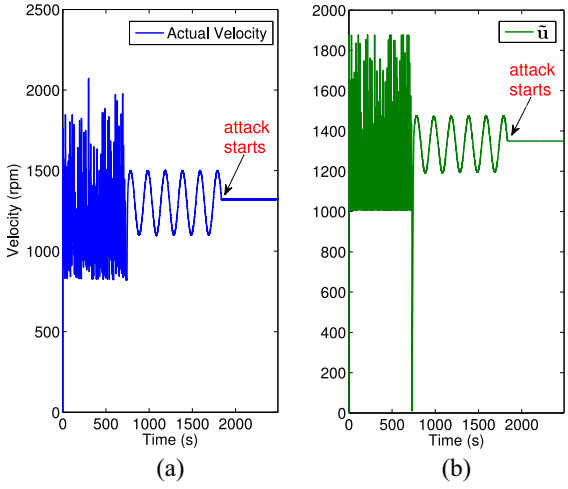

Fig. 6. Data recorded by the client computer, including the (a) actual motor velocity and (b) control command.

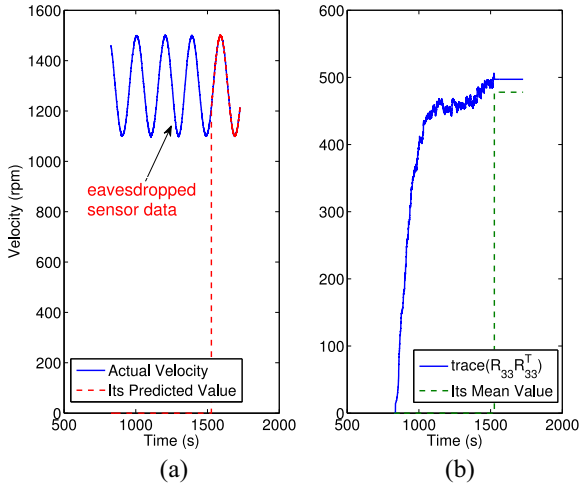

Fig. 7. Data recorded by the attacker's computer, including the (a) actual motor velocity and its predicted value and (b) trace of $R_{33}R_{33}^T$ and its mean value.

mean and variance of $y[k]-\acute{y}[k]$ are calculated. Besides, from Fig. 7(a), it is clear that the motor velocities have been predicted well. On this basis, at the 1836th second, the first packet including the forged sensor information $\ddot{y}[k]$ is sent to the server computer by the attacker's computer. After that, every 0.2 s, a false speed value fabricated by capturing the control command coming from the TCP server is given to the server computer. As a result, the normal communications between the TCP client and server are blocked. Fig. 6(a) and (b) indicates that, after the attack occurs, no control instruction can arrive at the client computer through the network, and therefore the motor velocities are held nearly constant except small variations derived from external disturbance, e.g., power fluctuations. From Fig. 5(b), it is observed that J_k is always less than the threshold value J_{th}, which proves that the attack launched in this experiment is a stealthy attack. Furthermore, Fig. 4(a) shows that there is no difference between the appearance of the red solid line before and after the attack is started. Nevertheless, the data received by the server computer after the 1836th second are actually forged by the adversary.

REFERENCES

[1] H. Ye, "Security protection technology of cyber-physical systems," *Int. J. Security Appl.*, vol. 9, no 2, pp. 159–168, 2015.

[2] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, Jan. 2015.

[3] Z.-H. Pang, G.-P. Liu, D. H. Zhou, F. Y. Hou, and D. H. Sun, "Two-channel false data injection attacks against output tracking control of networked systems," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3242–3251, May 2016.

[4] D. R. Ding, G. L. Wei, S. J. Zhang, Y. R. Liu, and F. E. Alsaadi, "On scheduling of deception attacks for discrete-time networked systems equipped with attack detectors," *Neurocomputing*, vol. 219, pp. 99–106, Jan. 2017.

[5] Y. Mo and B. Sinopoli, "On the performance degradation of cyber-physical systems under stealthy integrity attacks," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2618–2624, Sep. 2016.

[6] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proc. ACM Conf. Comput. Commun. Security*, 2009, pp. 21–32.

[7] G. Dán and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, 2010, pp. 214–219.

[8] Q. Y. Yang *et al.*, "On false data-injection attacks against power system state estimation: Modeling and countermeasures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 717–729, Mar. 2014.

[9] L. Xie, Y. Mo, and B. Sinopoli, "Integrity data attacks in power market operations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 659–666, Dec. 2011.

[10] O. Kosut, L. Jia, and R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec. 2011.

[11] M. Esmalifalak, Z. Han, and L. Song, "Effect of stealthy bad data injection on network congestion in market based power system," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2012, pp. 2468–2472.

[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] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Proc. 47th Annu. Allerton Conf. Commun. Control Comput.*, Oct. 2009, pp. 911–918.

[14] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on SCADA systems," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 4, pp. 1396–1407, Jul. 2014.

[15] Y. Huang *et al.*, "Bad data injection in smart grid: Attack and defense mechanisms," *IEEE Commun. Mag.*, vol. 51, no. 1, pp. 27–33, Jan. 2013.

[16] J. Kim, L. Tong, and R. J. Thomas, "Subspace methods for data attack on state estimation: A data driven approach," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1102–1114, Mar. 2015.

[17] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Inf. Sci.*, vol. 235, pp. 3–35, Jun. 2013.

[18] J.-S. Wang and G.-H. Yang, "Data-driven output-feedback fault-tolerant control for unknown dynamic systems with faults changing system dynamics," *J. Process Control*, vol. 43, pp. 10–23, Jul. 2016.

[19] J.-S. Wang and G.-H. Yang, "Data-driven output-feedback fault-tolerant compensation control for digital PID control systems with unknown dynamics," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7029–7039, Nov. 2016.

[20] J.-S. Wang and G.-H. Yang, "Data-driven output-feedback fault-tolerant tracking control method and its application to a DC servo motor system with unknown dynamics," *IEEE/ASME Trans. Mechatronics*, to be published.

[21] J.-S. Wang and G.-H. Yang, "Output-feedback control of unknown linear discrete-time systems with stochastic measurement and process noise via approximate dynamic programming," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2017.2726004.

[22] J.-S. Wang and G.-H. Yang, "Data-driven compensation method for sensor drift faults in digital PID systems with unknown dynamics," *J. Process Control*, vol. 65, pp. 15–33, May 2018.

[23] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input–output measured data," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2770–2779, Dec. 2015.

[24] B. Huang, S. X. Ding, and S. J. Qin, "Closed-loop subspace identification: An orthogonal projection approach," *J. Process Control*, vol. 15, no. 1, pp. 53–66, Feb. 2005.

[25] S. X. Ding, "Data-driven design of monitoring and diagnosis systems for dynamic processes: A review of subspace technique based schemes and some recent results," *J. Process Control*, vol. 24, no. 2, pp. 431–449, Feb. 2014.

[26] R. Hallouzi and M. Verhaegen, "Reconfigurable fault tolerant control of a Boeing 747 using subspace predictive control," presented at the AIAA Guid. Navig. Control Conf. Exhibit, Hilton Head, SC, USA, Aug. 2007, pp. 1–18.

[27] D. Xu, J. Liu, X.-G. Yan, and W. Yan, "A novel adaptive neural network constrained control for a multi-area interconnected power system with hybrid energy storage," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6625–6634, Aug. 2018.

[28] D. Liu and G.-H. Yang, "Data-driven adaptive sliding mode control of nonlinear discrete-time systems with prescribed performance," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2779564.

[29] D. Xu, Y. Shi, and Z. Ji, "Model-free adaptive discrete-time integral sliding-mode-constrained-control for autonomous 4WMV parking systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 834–843, Jan. 2018.

[30] M. Sveda and R. Vrba, "Cyber-physical systems networking with TCP/IP: A security application approach," in *Proc. IEEE AFRICON*, 2013, pp. 101–106.

[31] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ., 2012.

[32] S. Boyd, L. E. Ghaoui, E. Fern, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA, USA: Soc. Ind. Appl. Math., 1994.

[33] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*. London, U.K.: Imperial College Press, 2009.

[34] J. Wang and S. J. Qin, "Closed-loop subspace identification using the parity space," *Automatica*, vol. 42, no. 2, pp. 315–320, Feb. 2006.

Jun-Sheng Wang received the B.E. degree in automation from Northeastern University, Shenyang, China, in 2005, where he is currently pursuing the Ph.D. degree in control science and engineering with the College of Information Science and Engineering.

His current research interests include cyber-physical systems, fault-tolerant control, data-driven controller design and optimization, and reinforcement learning techniques and their application in industrial processes.

Guang-Hong Yang (SM'04) received the B.S. and M.S. degrees in mathematics and the Ph.D. degree in control theory and control engineering from Northeast University, Shenyang, China, in 1983, 1986, and 1994, respectively.

From 2001 to 2005, he was a Research Scientist/Senior Research Scientist with the National University of Singapore, Singapore. He is currently a Professor and the Dean with the College of Information Science and Engineering, Northeastern University. His current research interests include fault-tolerant control, fault detection and isolation, cyber-physical systems, and robust control.

Dr. Yang is a Deputy Editor-in-Chief for the *Journal of Control and Decision*, an Editor for the *International Journal of Control, Automation and Systems*, and an Associate Editor for the *International Journal of Systems Science*, the *IET Control Theory and Applications* and the IEEE TRANSACTIONS ON FUZZY SYSTEMS.