TU Delft

Delft University of Technology

# Deep reinforcement learning control approach to mitigating actuator attacks

Wu, Chengwei; Pan, Wei; Staa, Rick; Liu, Jianxing; Sun, Guanghui; Wu, Ligang

**DOI**
10.1016/j.automatica.2023.110999

**Publication date**
2023

**Document Version**
Final published version

**Published in**
Automatica

**Citation (APA)**
Wu, C., Pan, W., Staa, R., Liu, J., Sun, G., & Wu, L. (2023). Deep reinforcement learning control approach to mitigating actuator attacks. *Automatica*, *152*, Article 110999. https://doi.org/10.1016/j.automatica.2023.110999

**Important note**
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

**Copyright**
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

*This work is downloaded from Delft University of Technology.*
*For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.*

*Green Open Access added to TU Delft Institutional Repository*

*'You share, we take care!' - Taverne project*

*https://www.openaccess.nl/en/you-share-we-take-care*

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Automatica 152 (2023) 110999

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

# Deep reinforcement learning control approach to mitigating actuator attacks☆

Chengwei Wu [a], Wei Pan [b,c], Rick Staa [c], Jianxing Liu [a], Guanghui Sun [a], Ligang Wu [a,*]

[a] Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, PR China
[b] Department of Computer Science, The University of Manchester, United Kingdom
[c] Department of Cognitive Robotics, Delft University of Technology, Netherlands

ARTICLE INFO

*Article history:*
Received 18 July 2021
Received in revised form 10 July 2022
Accepted 27 February 2023
Available online 31 March 2023

*Keywords:*
Cyber–physical systems
False data injection attacks
Deep reinforcement learning
Lyapunov stability

ABSTRACT

This paper investigates the deep reinforcement learning based secure control problem for cyber–physical systems (CPS) under false data injection attacks. We describe the CPS under attacks as a Markov decision process (MDP), based on which the secure controller design for CPS under attacks is formulated as an action policy learning using data. Rendering the soft actor–critic learning algorithm, a Lyapunov-based soft actor–critic learning algorithm is proposed to offline train a secure policy for CPS under attacks. Different from the existing results, not only the convergence of the learning algorithm but the stability of the system using the learned policy is proved, which is quite important for security and stability-critical applications. Finally, both a satellite attitude control system and a robot arm system are used to show the effectiveness of the proposed scheme, and comparisons between the proposed learning algorithm and the classical PD controller are also provided to demonstrate the advantages of the control algorithm designed in this paper.

© 2023 Elsevier Ltd. All rights reserved.

## 1. Introduction

Cyber–physical systems (CPS) can characterize the interactions between the physical layer and the cyber space (Lee, 2008). Along with the increasing development of communication, computer, and control, CPS will be found everywhere in the future, examples of which are the smart grid, automation vehicles, transportation, process control systems, etc. However, due to interactions between the physical layer and the cyber space, adversaries can hijack into the cyber space and deteriorate the physical system (Cardenas et al., 2009). Several cyber attack events have been reported such as the Maroochy water services (Abrams & Weiss, 2008), the Stuxnet (Farwell & Rohozinski, 2011), and the Ukraine blackout (Liang, Weller, Zhao, Luo, & Dong, 2016). Therefore, how to design secure schemes for CPS under attacks has been an active yet challenging topic and considerable results have been proposed by using different discipline knowledge (Chen, Touati, & Zhu, 2019; Giraldo et al., 2018; Wu, Li, Pan, Liu, & Wu, 2021).

Researchers in the control community have been also dedicated to the security problem of CPS under attacks. Results on the modeling of attacks (Teixeira, Pérez, Sandberg, & Johansson, 2012), secure control (Jin, Haddad, & Yucelen, 2017), secure estimation (Fawzi, Tabuada, & Diggavi, 2014), optimal attack strategy design and power allocation (Guo, Shi, Johansson, & Shi, 2016) can be found in the literature. Generally speaking, attacks can be classified into two types, namely, the denial-of-service (DoS) attack and the deception attacks (for example, false data injection attack, zero dynamics attack) (Teixeira et al., 2012). For the DoS attack, it can be executed without knowing the system knowledge, which makes it to be a common attack. To make CPS under DoS attacks to preserve the desired performance, several remarkable results have been proposed, see for example De Persis and Tesi (2015), Wu, Wu, Liu, and Jiang (2019) and the references therein. Due to the limited communication resource, a traditional time-based communication scheme is not good enough for CPS. A reasonable communication scheme should be designed to exchange information. Some event-triggered secure control schemes have been designed for CPS under DoS attacks to maintain the system performance and save the limited communication resource (Dolk, Tesi, De Persis, & Heemels, 2016; Feng & Hu, 2019). Different from the above results, researchers have investigated how to

☆ This work was supported in part by the National Natural Science Foundation of China under Grant 62033005, Grant 62203136, Grant 62022030, Grant 62173107, in part by the Natural Science Foundation of Heilongjiang Province, PR China under Grant ZD2021F001, in part by the Key Research and Development Program of Heilongjiang Province, PR China under Grant 2022ZX01A18. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor James Lam under the direction of Editor Ian R. Petersen.

* Corresponding author.
*E-mail addresses:* chengweiwu@hit.edu.cn (C. Wu), wei.pan@manchester.ac.uk (W. Pan), rick.staa@outlook.com (R. Staa), jx.liu@hit.edu.cn (J. Liu), guanghuisun@hit.edu.cn (G. Sun), ligangwu@hit.edu.cn (L. Wu).

https://doi.org/10.1016/j.automatica.2023.110999
0005-1098/© 2023 Elsevier Ltd. All rights reserved.

design optimal DoS attack sequence and allocate attack power from the attacker's perspective (Qin, Li, Shi, & Yu, 2018; Zhang & Zheng, 2018), based on which the system designer can improve its defending scheme. Compared with DoS attacks, deception attacks, which are constructed by using the system knowledge are much stealthier. Great progress has been made for the security problem of CPS under deception attacks. For example, in Fawzi et al. (2014), the secure estimation problem under sparse attacks has been solved, and the upper bound of attacked sensors has been derived. Based on the conclusion in Fawzi et al. (2014), several improved results have been proposed (Li, Zhou, Li, Li, & Lu, 2019; Lu & Yang, 2019; Wu, Hu, Liu and Wu, 2018). The authors of Mo and Sinopoli (2015) analyzed the system degradation of CPS under deception attacks. An adaptive control framework has been proposed to mitigate false data injection attacks in Jin et al. (2017). Using the moving target defending scheme, a proactive secure control algorithm has been proposed in Wu et al. (2022). The authors of Ding, Han, Wang, and Ge (2019) reviewed recent results on the model-based secure control and estimation of CPS. Although effective schemes have been proposed, all these results rely on the physical model. Also, the linear model has been used in existing results yet such a simple model cannot characterize CPS under attacks precisely. Reinforcement learning, which solely uses data to design systems can get rid of the system model (Sutton & Barto, 2018). Therefore, it is an alternative way to design secure control schemes for CPS under attacks.

Reinforcement learning is gaining more and more attention. To make such an approach available to continuous-control systems, deep neural networks have been introduced to approximate the policy to be learned and the Q-function, based on which the deep reinforcement learning technique has been proposed (Mnih et al., 2015). Several representative deep reinforcement learning algorithms have been widely applied to robotic control and atari games, for example, the TRPO algorithm (Schulman, Levine, Abbeel, Jordan, & Moritz, 2015), the PPO algorithm (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017), the DDPG algorithm (Lillicrap et al., 2015), and the SAC algorithm (Haarnoja, Zhou, Abbeel, & Levine, 2018). Although the deep reinforcement learning algorithm can perform well, only the convergence of the learning algorithm is proved. The stability, which is a fundamental problem for a control system is not guaranteed in the deep reinforcement learning framework. For stability-critical systems, such learning algorithms cannot be deployed due to the absence of a stability guarantee. In the control community, Lyapunov function is often used to analyze the system stability. The authors of Chow, Nachum, Duenez-Guzman, and Ghavamzadeh (2018), Perkins and Barto (2001, 2002) have introduced the Lyapunov function in the reinforcement learning algorithm. But it is used to guarantee the safety of the agent in the training instead of the stability. To guarantee stability, Zhang et al. in Zhang, Dong and Pan (2020), Zhang, Pan and Reppa (2020) respectively proposed basic control based SAC algorithms for ships and multi-agents. Furthermore, Lyapunov-based soft actor–critic algorithms have been proposed for traditional control and estimation design in our previous work, in which the stability has been proved by solely using data (Han, Zhang, Wang, & Pan, 2020; Hu, Wu, & Pan, 2020) and a learned Lyapunov function constraint. Nevertheless, there exist few relative results concerning designing a secure control scheme for CPS against attacks by using deep reinforcement learning, which motivates this work.

In this paper, the deep reinforcement learning based secure control problem of CPS under actuator attacks is first investigated. The proposed reinforcement learning approach is utilized to learn an appropriate control signal to mitigate attack effects. We do not utilize any machine learning approach/deep neural network to learn/identify/reconstruct the false data injection attack signal to be used as compensation. When false data injection attacks occur, system states can be affected. To maintain the states to be desired values, the secure control signal is learned by letting these states as input of actor network. Of course, machine learning, even some classic model-based control approaches that include interval observer, unknown input observer, and adaptive techniques can be utilized to identify attacks. We found in the literature that machine learning techniques are mostly used to detect and identify attacks. It should also be noted that attacks are different from faults. Attacks are designed arbitrarily under some constraints. If machine learning is used to identify attacks, it should be capable of dealing with the sudden change of attacks at a fast rate (often known as anomaly detection), which may be not easy to be guaranteed. Consequently, we employ deep reinforcement learning to directly learn the secure control signal based on the current states. The main contributions of this paper can be summarized as follows:

(1) It is the first time to develop a deep reinforcement learning secure control algorithm for CPS under actuator false data injection attacks. In our approach, the neural networks are directly employed to learn a secure controller based on the system states rather than estimate such attack as compensations to mitigate attacks.
(2) Compared with Haarnoja et al. (2018), Lillicrap et al. (2015), Schulman et al. (2015, 2017), the stability of the system using the learned policy is proved by solely using data in our paper, which is great progress. Unlike our previous results (Han et al., 2020; Hu et al., 2020), which give the proof of asymptotical stability, the exponential stability is proved in our paper.

The rest of the paper is organized as follows. In Section 2, the secure control problem of CPS under actuator attacks is formulated. In Section 3, the design and implementation of the deep reinforcement learning algorithm is presented. In Section 4, both the convergence of the proposed learning algorithm and the stability of CPS are analyzed. In Section 5, simulation results are given to show the effectiveness of the proposed scheme, and Section 6 concludes this paper.

*Notation:* The notations used throughout the paper are defined as follows. $\mathbb{R}_n$ denotes the $n$ dimensional Euclidean space. The superscripts "$T$" and "$-1$" respectively denote the matrix transpose and inverse. $\mathbb{E}\{x\}$ means the expectation of the stochastic variable $x$. diag$\{\cdot\}$ is a matrix with diagonal structure. $\|x\|$ denotes 2-norm of the vector.

## 2. System description and problem formulation

As shown in Fig. 1, components in CPS considered in this paper are distributed. Both the sensor–controller and controller–actuator sides are connected by open and shared communication networks. The adversary can monitor the system and obtain the system knowledge, based on which it constructs false data and injects them into the control signal to deteriorate the system performance. In this section, we mainly describe each module in a mathematical way, and formulate the problem to be solved in this paper.

### 2.1. Physical system description

In existing results, the physical system is often described by a linear model. Here, we do not impose limitations on the model, and we assume its dynamics can be described by the following general equation

$$\dot{x} = f(x, u), \tag{1a}$$

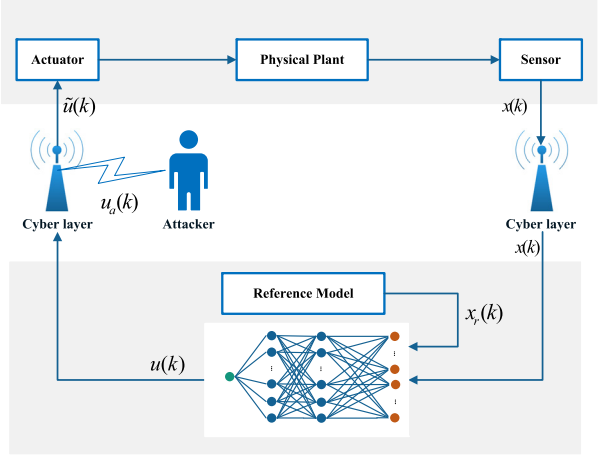


**Fig. 1.** Control framework of CPS under actuator false data injection attacks.

where $x \in \mathbb{R}_{n_x}$ is the system state vector, and $u \in \mathbb{R}_{n_u}$ denotes the control signal to be designed. In this paper, the system (1a) is controllable, except for which there exist no constraints on it.

In CPS, computer and communication networks are widely used. It is therefore necessary to discretize system (1a). Using the Euler approximation method (Gupta, 1995), the corresponding discrete-time dynamics can be described as

$$x(k+1) = (f\,(x(k), u(k)))\,\Delta t + x(k), \tag{1b}$$

where $\Delta t > 0$ is the sampling period.

According to Fig. 1, we know that a reference model is introduced to generate partial input of deep neural networks. Here, the reference model is a nominal system of (1b). In the reference model, the external disturbance, uncertainties, attacks are not included. It should be easy to design a controller for such a reference model with a good performance guarantee. For the reference model, its dynamical equation is as follows

$$x_r(k+1) = g\,(x_r(k), u_r(k))\,, \tag{2}$$

where $x_r(k) \in \mathbb{R}_{n_x}$ is the state of the reference model, and $u_r(k) \in \mathbb{R}_{n_u}$ is a well-designed controller, which is used to guarantee the stability of (2).

**Remark 1.** Since the system (2) is a nominal model of (1b), there exist considerable results, which can be used to design $u_r(k)$. For example, if system (2) is still in a nonlinear form, the sliding mode control, fuzzy control, and backstepping control can be introduced to design $u_r(k)$. Alternatively, a PID or PD controller can work. If system (2) is linear, an optimal controller can be readily obtained by solving a Riccati equation. The design process is omitted in our paper since it is easy to design it.

### 2.2. CPS under cyber attacks

In this paper, we assume that an adversary can construct false data and inject them into the control signal $u(k)$. The authors of Teixeira et al. (2012) have discussed how to describe different attacks and provided some attack models. Based on Teixeira et al. (2012), the control signal under attacks becomes

$$\tilde{u}(k) = u(k) + \Gamma u_a(k),$$

where $\tilde{u}(k)$ is the compromised control signal, $u_a(k)$ is the false data injection attack. $\Gamma$ denotes an attack distribution matrix. Here, $\Gamma$ is a diagonal matrix, the diagonal elements of which are 0 and 1. If the $i$th actuator is attacked, the $i$th diagonal element is 1, otherwise, it is 0.

For the attack signal $u_a(k)$, it is an unknown but bounded signal. It can be classified into two categories including the state-independent attack and the state-dependent attack (Yucelen, Haddad, & Feron, 2016).

(1) State-independent attacks
If $u_a(k)$ is time-invariant, $u_a(k) = \omega$ with $\omega$ being a constant. If $u_a(k)$ is time-varying, $u_a(k) = \omega(k) \leq \bar{\omega}$ with $\bar{\omega}$ being the upper bounded.

(2) State-dependent attacks
If $u_a(k)$ is time-invariant, $u_a(k)$ can be represented as $\omega x(k)$ with $\omega$ being a constant. If $u_a(k)$ is time-varying, $u_a(k)$ can be represented as $\omega(k)x(k)$ with $\omega(k)$ being a bounded time-varying function.

Considering the compromised control signal, the system (1b) is rewritten as

$$x(k+1) = \left(f\left(x(k), \tilde{u}(k)\right)\right)\Delta t + x(k). \tag{3}$$

Based on the above description, the objective of this paper is to learn a policy $\pi$ ($u(k)$ is sampled from $\pi$) for system (3) by using the deep reinforcement learning such that states of system (3) can exponentially converge in mean square, as described in Definition 1.

**Remark 2.** In this paper, a deep reinforcement learning algorithm will be proposed to learn the policy $\pi$. To improve the efficiency of training data, a basic controller can be added to $u(k)$, that is, $u(k) = u_{rl}(k) + u_b(k)$ with $u_{rl}(k)$ being the signal sampled from the policy $\pi$ and $u_b(k)$ being the basic controller. One can choose $u_r(k)$ as $u_b(k)$.

**Remark 3.** Considering that attacks can change suddenly, we do not utilize the neural network to directly identify/estimate/reconstruct false data injection attacks $u_a(k)$ as a compensation. The attack identifying/estimation/reconstruction scheme cannot immediately deal with the sudden change of attacks (Abbaspour, Sargolzaei, Forouzannezhad, Yen, & Sarwat, 2020). Before attacks are accurately identified/estimated/reconstructed, the stability/performance of CPS can be deteriorated even destroyed. In the following content, a deep reinforcement learning approach is designed to directly learn the secure controller $u_{rl}(k)$ based on the system states.

**Definition 1.** The state $x(k)$ is said to be exponentially stable in mean square if $\exists\ \eta > 0$ and $0 < \varphi < 1$, such that

$$\mathbb{E}[\|x(k)\|^2] \leq \eta \|x(0)\|^2 \varphi^k,$$

holds at all the time instants $k \geq 0$.

## 3. Deep reinforcement learning based secure controller design and implementation

This section mainly focuses on developing a deep reinforcement learning algorithm and discussing its implementation. We first formulate system (3) as an MDP. A Lyapunov-based soft actor–critic learning algorithm is proposed to learn a policy for the MDP, and a secure controller can be obtained by sampling the learned policy. Then, the implementation of the proposed learning is discussed by using deep neural networks. Next, they are introduced in detail.

### 3.1. Markov decision process

An MDP is described by a tuple including five elements, that is, $\langle \mathcal{S},\ \mathcal{A},\ \mathcal{P},\ \mathcal{C},\ \gamma \rangle$. Here, $\mathcal{S}$ means the state space, $\mathcal{A}$ denotes the action space, $\mathcal{P}$ is the transition probability distribution, $\mathcal{C}$ means the control cost, and $\gamma \in [0,\ 1)$ is the discount factor. Based on the above description, we know that the $x_r(k)$ is implicitly included in $u(k)$. Therefore, an MDP for system (3) is described as

$$\bar{x}(k+1) \sim \mathcal{P}\left(\bar{x}(k+1)|\bar{x}(k), u(k)\right), \tag{4}$$

where $\bar{x}(k)$ is the state with $\bar{x}(k) = [x(k),\ x_r(k)]$. $\mathcal{P}\left(\bar{x}(k+1)|\bar{x}(k),\ u(k)\right)$ means the transition probability from $\bar{x}(k)$ to $\bar{x}(k+1)$ under $u(k)$.

### 3.2. Reinforcement learning algorithm

In this section, an actor–critic reinforcement learning algorithm is developed to learn a policy for the MDP in (4). Different from existing results, which maximize the reward, we minimize the control cost in the design process. For the reference model, its performance will be guaranteed by a well-designed controller. If we can guarantee $x(k)$ exponentially converges to $x_r(k)$, the stability of system (3) can be preserved. Thus, the cost $\mathcal{C}(k)$ in this paper is defined as

$$\mathcal{C}(k) = \left(x(k) - x_r(k)\right)^T \left(x(k) - x_r(k)\right).$$

The objective of the reinforcement learning is to find an optimal policy to minimize the following state-value function

$$V_\pi\left(\bar{x}(k)\right) = \sum_{k}^{\infty} \sum_{u(k)} \pi\left(u(k)|\bar{x}(k)\right) \sum_{\bar{x}(k+1)} \mathcal{P}_{k+1|k} \times \left(\mathcal{C}(k) + \gamma V_\pi(\bar{x}(k+1))\right), \tag{5}$$

where $\mathcal{P}_{k+1|k} = \mathcal{P}\left(\bar{x}_{k+1}\,|\bar{x}(k), u(k)\right)$, $\pi$ is a policy to be learned. In reinforcement learning, $\pi$ is a Gaussian distribution, and $\pi\left(u(k)|\bar{x}(k)\right)$ can be obtained as

$$\pi\left(u(k)|\bar{x}(k)\right) = \mathcal{N}\left(u(k), \sigma\right), \tag{6}$$

where $\mathcal{N}\left(u(k), \sigma\right)$ denotes a Gaussian distribution with the mean value $u(k)$ and the covariance matrix $\sigma$, $\pi\left(u(k)|\bar{x}(k)\right)$ means the probability of choosing the action $u(k)$ at state $\bar{x}(k)$ from the policy $\pi$.

The Q-function $Q_\pi\left(\bar{x}(k), u(k)\right)$ instead of the state-value function is practically minimized to find an optimal policy $\pi$ in the training process. Based on (5) the Q-function is described as

$$Q_\pi\left(\bar{x}(k), u(k)\right) = \mathcal{C}(k) + \gamma \mathbb{E}_{\bar{x}(k+1)}\left[V_\pi(\bar{x}(k+1))\right], \tag{7}$$

where $\mathbb{E}_{\bar{x}(k+1)}[\cdot] = \sum_{\bar{x}(k+1)} \mathcal{P}_{k+1|k}[\cdot]$ is an expectation operator over the distribution of $\bar{x}(k+1)$.

To guarantee unknown action space is explored sufficiently, an entropy item can be added to the Q-function (7). Then, considering the entropy item, the Q-function (7) can be described as

$$Q_\pi\left(\bar{x}(k), u(k)\right) = \mathcal{C}(k) + \gamma \mathbb{E}_{\bar{x}(k+1)}\left[V_\pi(\bar{x}(k+1)) - \alpha \mathcal{H}\left(\pi\left(u(k+1)|\bar{x}(k+1)\right)\right)\right], \tag{8}$$

where $\alpha$ means a temperature parameter, which is used to adjust the relative importance of the entropy item. $\mathcal{H}\left(\pi\left(u(k+1)|\bar{x}(k+1)\right)\right)$ denotes the entropy of policy, and

$$\begin{aligned} &\mathcal{H}\left(\pi\left(u(k+1)|\bar{x}(k+1)\right)\right) \\ &= -\sum_{u(k)} \pi\left(u(k)|\bar{x}(k)\right) \ln\left(\pi\left(u(k)|\bar{x}(k)\right)\right) \\ &= -\mathbb{E}_\pi\left[\ln\left(\pi\left(u(k)|\bar{x}(k)\right)\right)\right]. \end{aligned}$$

**Remark 4.** In the reinforcement learning, the entropy item $\mathcal{H}\left(\pi\left(u(k+1)|\bar{x}(k+1)\right)\right)$ is added to explore unknown action space. More unknown space is explored, the better performance can be achieved. Thus, the added entropy item is $\mathcal{H}\left(\pi\left(u(k+1)|\bar{x}(k+1)\right)\right)$, that is, it should be maximized, for example the SAC algorithm. It is noted that the objective of our algorithm is to minimize the Q-function. To maximize the entropy item, $-\mathcal{H}\left(\pi\left(u(k+1)|\bar{x}(k+1)\right)\right)$ is added to the Q-function.

According to the above description, the reinforcement learning algorithm is to solve the following problem

$$\pi^* = \arg\min_{\pi \in \Pi} \left(\mathcal{C}(k) + \gamma \mathbb{E}_{\bar{x}(k+1)}\left[V_\pi(\bar{x}(k+1)) - \alpha \mathcal{H}\left(\pi\left(u(k+1)|\bar{x}(k+1)\right)\right)\right]\right), \tag{9}$$

where $\Pi$ is the policy.

By solving (9), we can obtain the optimal policy $\pi^*$, based on which $\pi^*\left(u(k)|\bar{x}(k)\right) = \mathcal{N}\left(u^*(k), \sigma^*\right)$ can be obtained. (9) is solved by using the reinforcement learning algorithm. When the training is completed, $\sigma^* = 0$, and the control signal $u^*(k)$ sampled from the learned policy $\pi^*$ is a deterministic mean value.

In the training process, two steps need to be executed repeatedly, that is, policy evaluation and policy improvement. In the policy evaluation step, a Bellman backup operation is applied to the Q-function with entropy item (8), namely,

$$\begin{aligned} &\mathcal{T}^\pi Q_\pi\left(\bar{x}(k), u(k)\right) \\ &= \mathcal{C}(k) + \gamma \mathbb{E}_{\bar{x}(k+1)}\left[\mathbb{E}_\pi\left[Q_\pi\left(\bar{x}(k+1), u(k+1)\right) - \alpha \mathcal{H}\left(\pi\left(u(k+1)|\bar{x}(k+1)\right)\right)\right]\right]. \end{aligned} \tag{10}$$

In the policy improvement step, the policy is updated by

$$\pi_{\text{new}} = \arg\min_{\pi' \in \Pi} \mathscr{D}_{\text{KL}}\left(\pi'(\cdot|x_k) \,\middle\|\, \frac{e^{\frac{-1}{\alpha} Q^{\pi_{\text{old}}}\left(\bar{x}(k), \cdot\right)}}{Z^{\pi_{\text{old}}}}\right), \tag{11}$$

where $\pi_{\text{old}}$ is the policy from the last update, $Q^{\pi_{\text{old}}}$ is the Q-value of $\pi_{\text{old}}$, $\mathscr{D}_{\text{KL}}$ means the Kullback–Leibler divergence, and $Z^{\pi_{\text{old}}}$ denotes a normalization factor. Then, (11) can be rewritten as

$$\pi^* = \arg\min_{\pi \in \Pi} \mathbb{E}_\pi\left[\alpha \ln\left(\pi\left(u(k)|\bar{x}(k)\right)\right) + Q\left(\bar{x}(k), u(k)\right)\right]. \tag{12}$$

### 3.3. Deep neural networks approximation

To implement the above reinforcement learning algorithm, deep neural networks constructed by fully connected multiple layer perceptrons are used to approximate the Q-function and the action.

In the constructed deep neural networks, the activation functions are often the rectified linear unit (ReLU) candidate defined as $\rho(z) = \max\{z, 0\}$ (Dahl, Sainath, & Hinton, 2013). To clearly show the construction of deep neural networks using multiple layer perceptrons, an example with two hidden layers is discussed. For the ReLU function, $\rho(z) = [\rho(z_1), \ldots, \rho(z_n)]^T$ when $z$ is a vector with $z = [z_1, \ldots, z_n]^T$. Then, the deep neural networks are constructed as

$$\underline{\text{DNN}}_{\text{w}}(z) = \text{w}_2 \left[\rho\left(\text{w}_1 \left[\rho\left(\text{w}_0 \begin{bmatrix} z \\ 1 \end{bmatrix}\right), 1\right]^T\right)^T, 1\right]^T,$$

where $\left[z^T,\ 1\right]^T$ with "1" being the bias, $\text{w} = [\text{w}_0, \text{w}_1, \text{w}_2]$ with $\text{w}_0$, $\text{w}_1$, and $\text{w}_2$ being the weighting coefficients of the neural networks to be trained.

As mentioned above, the constructed deep neural networks are respectively used to approximate the "critic" $Q_\pi(\bar{x}_k, u(k))$ and the "actor" $\pi(u(k)|\bar{x}(k))$. $\theta$ and $\phi$ are utilized to parameterize $Q(\bar{x}(k), u(k))$ and $\pi(u(k)|\bar{x}(k))$, namely, $Q_\theta(\bar{x}(k), u(k))$ and $\pi_\phi(u(k)|\bar{x}(k))$, which are shown in Fig. 2.

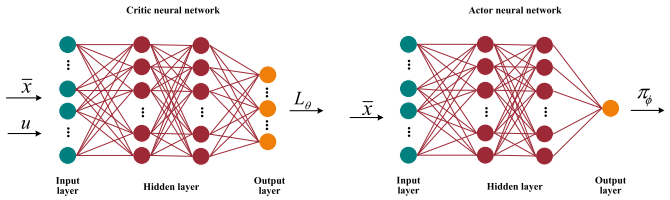


**Fig. 2.** Deep neural networks constructed for the "critic" and "actor".

In our paper, one objective is to guarantee the stability of CPS using the learned policy. Lyapunov theory is widely used to analyze the stability of systems. In the learning process, the Q-function $Q_\pi(\bar{x}_k, u(k))$ is regarded as a Lyapunov function $\mathcal{L}(k)$, and the critic neural network is used to approximate the Lyapunov function $\mathcal{L}(k)$. If we can learn a Lyapunov function, which satisfies the requirement of stability, the stability of CPS using the learned policy can be guaranteed. In the following contents, we will use the Lyapunov function $\mathcal{L}(k)$ to replace the Q-function $Q_\pi(\bar{x}_k, u(k))$.

$\mathcal{L}_\theta(k)$ means the parameterized Lyapunov function with $\mathcal{L}_\theta(k) = \underline{\text{DNN}}_\theta(\bar{x}(k), u(k))$. For the actor neural network, its output includes two parts, that is, the parameterized secure controller $u_\phi(k)$ and the standard deviation $\sigma_\phi$. According to (6), $\pi_\phi(u(k)|\bar{x}(k)) = \underline{\text{DNN}}_\phi\left(u_\phi(k), \sigma_\phi^2\right)$.

### 3.4. Implementation of Lyapunov-based soft actor–critic deep reinforcement learning control algorithm

In the above sections, we described the reinforcement learning algorithm, the construction of deep neural networks, and the definition of the Lyapunov function used in the training process. This section mainly discusses how to implement a soft actor–critic deep reinforcement learning control algorithm with Lyapunov function constraints. Fig. 3 shows the training process of the proposed algorithm. As shown in Fig. 3, systems in (2) and (3) run to generate training data, which is restored in the replay memory $\mathcal{M}$. By randomly sampling a batch of collected data from the memory $\mathcal{M}$, the policy evaluation and policy improvement steps are executed repeatedly. Then, the improved policy $\pi_\phi(u(k)|\bar{x}(k))$ is applied to the system (3) to generate data for training.

In the policy evaluation step, $\pi_\phi(u(k)|\bar{x}(k))$ should minimize the following Bellman residual equation

$$\mathcal{J}_\mathcal{L}(\theta) = \mathbb{E}_{(\bar{x}_k, u(k)\sim\mathcal{M})}\left\{\frac{1}{2}\left(\mathcal{L}_\theta(k) - \mathcal{L}_{\text{target}}\right)^2\right\}, \tag{13}$$

where $(\bar{x}(k), u(k) \sim \mathcal{M})$ denotes the random sample of $(\bar{x}(k), u(k))$ from $\mathcal{M}$, and

$$\mathcal{L}_{\text{target}} = \mathcal{C}(k) + \gamma\mathbb{E}_{\bar{x}(k+1)}\left[\mathbb{E}_\pi\left[\mathcal{L}_{\bar{\theta}}(k+1) + \alpha\ln(\pi_\phi)\right]\right],$$

with $\bar{\theta}$ being the target parameter.

The ADAM-optimizer (Kingma & Ba, 2014) is used to minimize the residual equation in the training process. For a batch of data with size $|\mathcal{B}|$, the stochastic gradient of (13) is

$$\nabla_\theta\mathcal{J}_\mathcal{L}(\theta) = \sum\frac{\nabla_\theta\mathcal{L}_\theta}{|\mathcal{B}|}\left(\mathcal{L}_\theta(k) - \mathcal{L}_{\text{target}}\right).$$

Different from existing results (Haarnoja et al., 2018; Lillicrap et al., 2015; Schulman et al., 2015, 2017), the stability of the system using the learned policy will be guaranteed in our paper. Therefore, the property of a Lyapunov function should be preserved when the policy is updated. Namely, a Lyapunov constraint should be added to the problem in (12). Then, (12) can be rewritten as

$$\pi_{\text{new}} = \arg\min_{\pi\in\Pi}\mathbb{E}\left[\alpha\ln(\pi(u(k)|\bar{x}(k))) + \mathcal{L}(k)\right]$$

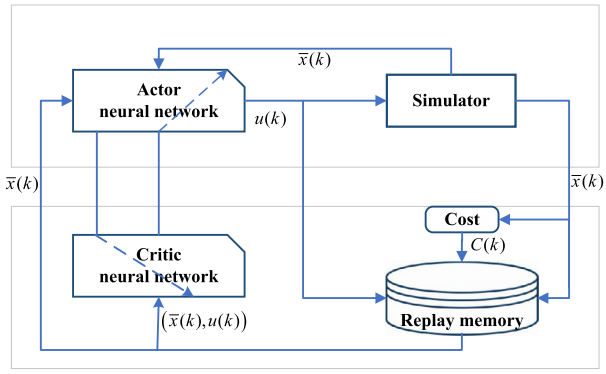


**Fig. 3.** Offline training process of the proposed learning algorithm. Here, the simulator module includes the system (3) and the reference model (2).

$$\text{s.t. } \Delta\mathcal{L}(k) < 0, \tag{14}$$

where $\Delta\mathcal{L}(k) = \mathcal{L}_\theta(k+1) - \mathcal{L}(k) + \beta\mathcal{C}(k)$.

To solve the constrained optimization problem (14), the Lagrangian multiplier is introduced. (14) is further described as

$$\pi_{\text{new}} = \arg\min_{\pi\in\Pi}\mathbb{E}\left[\alpha\ln(\pi(u(k)|\bar{x}_k)) + \mathcal{L}(k) + \lambda\Delta\mathcal{L}(k)\right], \tag{15}$$

where $\lambda$ is a Lagrangian multiplier.

Then, the objective of the policy improvement is to minimize the following equation

$$\mathcal{J}_\pi(\phi) = \mathbb{E}_{(\bar{x}_k, u(k)\sim\mathcal{M})}\left[\alpha\ln(\pi_\phi) + \mathcal{L}_\theta(k) + \lambda\Delta\mathcal{L}_\theta(k)\right],$$

which is equivalent to

$$\mathcal{J}_\pi(\phi) = \mathbb{E}_{(\bar{x}_k, u(k)\sim\mathcal{M})}\left[\alpha\ln(\pi_\phi) + \lambda\Delta\mathcal{L}_\theta(k)\right].$$

The gradient of the above equation w.r.t. $\phi$ is derived as

$$\nabla_\phi\mathcal{J}_\pi(\phi) = \sum\frac{\Psi}{|\mathcal{B}|},$$

where $\Psi = \alpha\nabla_{u(k)}\ln\pi_\phi\nabla_\phi u_\phi(k) + \alpha\nabla_\phi\ln\pi_\phi + \lambda\nabla_{u(k+1)}\mathcal{L}_\theta(k+1)\nabla_\phi\pi_\phi(\cdot|\bar{x}(k+1))$.

The temperature parameter $\alpha$ is updated by minimizing the following function (Haarnoja et al., 2018)

$$\mathcal{J}_\alpha = \mathbb{E}_\pi\left\{-\alpha\ln\pi(u(k)|\bar{x}(k)) - \alpha\mathcal{H}\right\},$$

where $\mathcal{H}$ denotes a target entropy.

For the Lagrangian multiplier $\lambda$, it is learned by maximizing

$$\mathcal{J}(\lambda) = \lambda\mathbb{E}\left[\mathcal{L}_\theta(k+1) - \mathcal{L}(k) + \beta\mathbb{E}_{\bar{x}(k)\sim\mu_\pi}\left[\mathcal{C}(k)\right]\right].$$

According to the above discussion, the implementation of the Lyapunov-based soft actor–critic deep reinforcement learning control algorithm is summarized in Algorithm 1. By using such an algorithm, a policy $\pi^*$ satisfying the Gaussian distribution can be learned. During the inference, a mean value $u(k)$ can be applied to stabilize CPS under attacks.

**Remark 5.** Readers may be curious about the imitation of attacks in the training process. In practical applications, there exist physical constraints on actuators, for example, the saturation. Also, the adversary constructs false data injection based on the system knowledge, and adversaries intend to make attacks to be much stealthier. Thus, an adversary will not inject very large signals into the actuator. Besides, several researchers have studied how to design optimal stealthy attacks to bypass the anomaly detection and deteriorate the system performance (Teixeira, 2019; Zhang & Venkitasubramaniam, 2017). In the training, we can use a uniform

distribution with lower and upper bounds to generate attack signals. Alternatively, an optimal attack policy can be given by using the existing results. Neural networks can deal with some uncertainties. Although exact attack signals are unavailable in the training process, the learned policy can perform well if neural networks are trained by enough data.

**Remark 6.** The anomaly detection and estimate scheme is not designed in this paper. As can be seen from our previous work (Hu et al., 2020), a learning based filter can be designed with the bounded estimate errors guarantee. In the future, a filter-based attack detection and secure control scheme will be co-designed solely using data, using which the security of CPS can be further improved.

**Algorithm 1** Lyapunov-Based Reinforcement Learning Control Algorithm

Design a controller for the reference model (2)
Set initial values for temperature parameter $\alpha$, Lagrangian multiplier $\lambda$, and the learning rates $\iota_{\mathcal{L}}$, $\iota_{\pi}$, $\iota_{\alpha}$, $\iota_{\lambda}$
Initialize $\theta$ for $\mathcal{L}_\theta$, $\phi$ for $\pi_\phi$, and the replay memory $\mathcal{M}$
Set the target parameter $\bar{\theta}$ as $\bar{\theta} \leftarrow \theta$
**while** Training **do**
  **for** each data collection step **do**
    Sample $u(k)$ from the policy $\pi_\theta(\bar{x}_k|u(k))$
    Apply $u(k)$ to system (3)
    Run system (3) and the reference model (2) to generate data $\bar{x}(k)$
    Update the memory $\mathcal{M} \leftarrow \mathcal{M} \cup \bar{x}_k$
  **end for**
  **for** each gradient step **do**
    $\theta \leftarrow \theta - \iota_{\mathcal{L}} \nabla_\theta \mathcal{J}_{\mathcal{L}}(\theta)$,
    $\phi \leftarrow \phi - \iota_{\pi} \nabla_\phi \mathcal{J}_{\pi}(\phi)$
    $\alpha \leftarrow \alpha - \iota_{\alpha} \nabla_\alpha \mathcal{J}_{\alpha}(\alpha)$
    $\lambda \leftarrow \lambda - \iota_{\lambda} \nabla_\lambda \mathcal{J}_{\lambda}(\lambda)$
    $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$,
  **end for**
**end while**
Output optimal parameters $\theta^*$, $\phi^*$, $\lambda^*$, and $\alpha^*$

## 4. Convergence and stability analysis

This section includes two parts, that is, the convergence analysis of Algorithm 1 and the stability analysis of the system (3). For the convergence analysis of Algorithm 1, it can be completed by referring to Haarnoja et al. (2018). As to the stability analysis of the system (3), it is the main contribution of our paper. By using the Lyapunov theory, Lebesgue's Dominated convergence theorem and some other techniques, the stability of the system using the learned policy is proved. Next, we give the details.

### 4.1. Algorithm convergence analysis

Algorithm 1 repeatedly executes the policy evaluation and the policy improvement steps. Thus, its convergence is analyzed from these two aspects. Next, two lemmas are given for the two steps.

**Lemma 1** (*Haarnoja et al. (2018) Policy Evaluation*)**.** *Considering the Bellman backup operation $\mathcal{T}^\pi$ in (10) and defining $\mathcal{L}^{i+1}(k) = \mathcal{T}^\pi \mathcal{L}^i(k)$, the sequence $\mathcal{L}^{i+1}(k)$ can converge to a soft value $\mathcal{L}^\pi$ of the policy $\pi$ as $i \to \infty$.*

**Lemma 2** (*Policy Improvement*)**.** *Define $\pi_{old}$ as the last updated policy, and $\pi_{new}$ obtained from (14) as the new policy. For $\forall \bar{x}(k) \in \mathcal{S}$ and $\forall u(k) \in \mathcal{A}$, there always exists $\mathcal{L}^{\pi_{new}}(k) \le \mathcal{L}^{\pi_{old}}(k)$.*

**Proof.** Based on (14), the following inequality can be derived

$$\begin{aligned}&\mathbb{E}_{\pi_{\text{new}}}\left[\alpha \ln(\pi_{\text{new}}(u(k)|\bar{x}_k)) + \mathcal{L}_{\pi_{\text{old}}}(k)\right]\\ &\le \mathbb{E}_{\pi_{\text{old}}}\left[\alpha \ln(\pi_{\text{old}}(u(k)|\bar{x}_k)) + \mathcal{L}_{\pi_{\text{old}}}(k)\right],\end{aligned}$$

which implies

$$\mathbb{E}_{\pi_{\text{new}}}\left[\mathcal{L}_{\pi_{\text{old}}} + \alpha \ln(\pi_{\text{new}}(u(k)|\bar{x}(k)))\right] \le V_{\pi_{\text{old}}}(\bar{x}(k)).$$

Then, the following inequality can be obtained

$$\begin{aligned}\mathcal{L}_{\pi_{\text{old}}}(k) &= \mathcal{C}(k) + \gamma \mathbb{E}_{\bar{x}(k+1)}\left[V_{\pi_{\text{old}}}(\bar{x}(k))\right]\\ &\ge \mathcal{C}(k) + \gamma \mathbb{E}_{\bar{x}(k+1)}\left[\mathbb{E}_{\pi_{\text{new}}}\left[\right.\right.\\ &\quad \mathcal{L}_{\pi_{\text{old}}}(k+1)\\ &\quad \left.\left.-\alpha \ln(\pi_{\text{new}}(u(k+1)|\bar{x}(k+1)))\right]\right]\\ &\quad \vdots\\ &\ge \mathcal{L}_{\pi_{\text{new}}}(k).\end{aligned}$$

The proof is completed. ■

Based on Lemmas 1 and 2, the following theorem is provided to show that the Lyapunov function $\mathbb{E}[\mathcal{L}(k)]$ can be learned after repeatedly executing the policy evaluation and policy improvement steps.

**Theorem 1.** *For $\forall \pi_0 \in \Pi$, the policy updated at ith policy improvement step $\pi_i$ ($i = 1, 2, \ldots, \infty$) can converge to optimal $\pi^*$, which ensures $\mathcal{L}_{\pi^*}(k) \le \mathcal{L}_{\pi_i}(k)$ holds for $\forall \bar{x}(k) \in \mathcal{S}$ and $u(k) \in \mathcal{A}$.*

**Proof.** According to Lemma 2, $\mathcal{L}(k)$ can be improved after executing the policy improvement step, that is $\mathcal{L}_{\pi_i}(k) \le \mathcal{L}_{\pi_{i-1}}(k)$. By repeatedly executing the policy evaluation and policy improvement, $\mathcal{L}_{\pi^*}(k) \le \mathcal{L}_{\pi_i}(k)$ holds for $\forall \pi_i \in \Pi$. ■

### 4.2. Data-based stability analysis

In Algorithm 1, we have learned a Lyapunov function $\mathcal{L}(k)$ as the critic function. The constraint $\Delta\mathcal{L}(k) = \mathcal{L}_\theta(k+1) - \mathcal{L}(k) + \beta \mathbb{E}_{\bar{x}(k)\sim\mu_\pi}[\mathcal{C}(k)] < 0$ is satisfied by executing the learning algorithm. Next, we will use the learned Lyapunov function to analyze the stability of the system in (3).

Before analyzing the stability, a common assumption in the reinforcement learning is given as follows.

**Assumption 1.** A Markov chain induced by a policy $\pi$ is ergodic with a unique distribution probability $q_\pi(\bar{x}(k))$ with $q_\pi(\bar{x}(k)) = \lim_{k\to\infty} \mathcal{P}(\bar{x}(k) \mid \rho, \pi, k)$.

**Theorem 2.** *If a Lyapunov function $\mathcal{L}(k)$ can be learned by Algorithm 1, and there exist constants $\tilde{\alpha}_1 > 0$, $\tilde{\alpha}_2 > 0$, $\beta \ge 0$ such that the following inequality holds*

$$\tilde{\alpha}_1 \mathcal{C}(k) \le \mathcal{L}(k) \le \tilde{\alpha}_2 \mathcal{C}(k), \tag{16}$$

$$\begin{aligned}&\mathbb{E}_{\bar{x}(k)\sim\mu_\pi}\left[\mathbb{E}_{\bar{x}(k+1)\sim\mathcal{P}_\pi}[\mathcal{L}(k+1)] - \mathcal{L}(k)\right]\\ &\le -\beta \mathbb{E}_{\bar{x}(k)\sim\mu_\pi}[\mathcal{C}(k)].\end{aligned} \tag{17}$$

*Then the system (3) is guaranteed to be exponentially stable in mean square, i.e.,*

$$\mathbb{E}_{\bar{x}(k)\sim\mu_\pi}[x(k)] \le \sigma^k \frac{\tilde{\alpha}_2}{\tilde{\alpha}_1} \mathbb{E}_{\bar{x}(0)\sim\mu_\pi}[x(0)], \tag{18}$$

*where*

$$\mu_\pi(\bar{x}(k)) \triangleq \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N} \mathcal{P}(\bar{x}(k) \mid \rho, \pi, k)$$

*is the state distribution, and $\sigma \in (0, 1)$.*

**Proof.** Based on Assumption 1, the sampling distribution $\mu_\pi(\bar{x}(k))$ exists. When $k$ goes to $\infty$, $q_\pi(\bar{x}(k)) = \mathcal{P}(\bar{x}(k) \mid \rho, \pi, k)$. Using the Abelian theorem, the sequence $\left\{\frac{1}{N}\sum_{k=0}^{N} P(\bar{x}(k) \mid \rho, \pi, k), N \in \mathbb{Z}_+\right\}$ also converges, and $\mu_\pi(\bar{x}(k)) = q_\pi(\bar{x}(k))$. According to the above discussion, (17) is rewritten as

$$\int_{\mathcal{S}} \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N} P(\bar{x}(k)|\rho, \pi, k) \left(\mathbb{E}_{\mathcal{P}_\pi(\bar{x}(k+1)|\bar{x}(k))} [\mathcal{L}(k+1)] - \mathcal{L}(k)\right) \mathrm{d}\bar{x}(k) \leq -\beta \mathbb{E}_{\bar{x}(k)\sim q_\pi} \|\bar{x}(k)\|^2. \tag{19}$$

Based on the Lebesgue's Dominated convergence theorem (Royden, 1968), if a sequence $f_n(\bar{x}(k))$ converges point-wise to a function $f$ and is dominated by some integrable function $g(\bar{x}(k))$ in the sense that,

$$|f_n(\bar{x}(k))| \leq g(\bar{x}(k)), \forall \bar{x}(k) \in \mathcal{S}, \forall n.$$

Then, the following equation can be obtained

$$\lim_{n\to\infty} \int_{\mathcal{S}} f_n(\bar{x}(k)) \mathrm{d}\bar{x}(k) = \int_{\mathcal{S}} \lim_{n\to\infty} f_n(\bar{x}(k)) \mathrm{d}\bar{x}(k).$$

Using the above equation, (19) can be described as

$$\begin{aligned}
&\int_{\mathcal{S}} \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N} \mathcal{P}(\bar{x}(k) \mid \rho, \pi, k) \left( \int_{\mathcal{S}} \mathcal{P}_\pi \right. \\
&\left. (\bar{x}(k+1)|\bar{x}(k)) \, \mathcal{L}(k+1) \mathrm{d}\bar{x}(k+1) - \mathcal{L}(k)) \, \mathrm{d}\bar{x}(k) \right. \\
&= \lim_{N\to\infty} \frac{1}{N} \left( \sum_{k=1}^{N+1} \mathbb{E}_{\mathcal{P}(\bar{x}(k)|\rho,\pi,k)}[\mathcal{L}(k)] \right. \\
&\left. - \sum_{k=0}^{N} \mathbb{E}_{\mathcal{P}(\bar{x}(k)|\rho,\pi,k)}[\mathcal{L}(k)] \right) \\
&= \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N} \left( \mathbb{E}_{\mathcal{P}(\bar{x}(k+1)|\rho,\pi,k+1)}[\mathcal{L}(k+1)] \right. \\
&\left. - \mathbb{E}_{\mathcal{P}(\bar{x}(k)|\rho,\pi,k)}[\mathcal{L}_k] \right).
\end{aligned} \tag{20}$$

There always exists a scalar $\sigma$ such that the following equation holds

$$\left(\frac{1}{\sigma} - 1\right) \tilde{\alpha}_2 - \frac{\beta}{\sigma} = 0.$$

Combining the above equation and the condition in (16) yields

$$\begin{aligned}
&\frac{1}{\sigma^{\iota+1}} \mathbb{E}_{\mathcal{P}(\bar{x}_{\iota+1}|\rho,\pi,\iota+1)}[\mathcal{L}(\iota+1)] - \frac{1}{\sigma^{\iota}} \mathbb{E}_{\mathcal{P}(\bar{x}_{\iota}|\rho,\pi,\iota)}[\mathcal{L}(\iota)] \\
&= \frac{1}{\sigma^{\iota+1}} \left( \mathbb{E}_{\mathcal{P}(\bar{x}(\iota+1)|\rho,\pi,\iota+1)}[\mathcal{L}(\iota+1)] - \mathbb{E}_{\mathcal{P}(\bar{x}(\iota)|\rho,\pi,\iota)}[\mathcal{L}(\iota)] \right) \\
&\quad + \frac{1}{\sigma^{\iota}} \left( \frac{1}{\sigma} - 1 \right) \mathbb{E}_{\mathcal{P}(\iota|\rho,\pi,\iota)}[\mathcal{L}(\iota)] \\
&\leq \frac{1}{\sigma^{\iota}} \left( -\frac{\beta}{\sigma} + (\frac{1}{\sigma} - 1)\tilde{\alpha}_2 \right) \mathcal{C}(k)
\end{aligned}$$

which implies

$$\frac{\mathbb{E}_{\mathcal{P}(\bar{x}(\iota+1)|\rho,\pi,\iota+1)}[\mathcal{L}(\iota+1)]}{\sigma^{\iota+1}} - \frac{\mathbb{E}_{\mathcal{P}(\bar{x}(\iota)|\rho,\pi,\iota)}[\mathcal{L}(\iota)]}{\sigma^{\iota}} \leq 0.$$

To sum the above inequality from $\iota = 0, 1, \ldots, k-1$ yields

$$\frac{1}{\sigma^k} \mathbb{E}_{\mathcal{P}(\bar{x}(k)|\rho,\pi,k)}[\mathcal{L}(k)] - \mathbb{E}_{\mathcal{P}(\bar{x}(0)|\rho,\pi,0)}[\mathcal{L}(0)] \leq 0,$$

which implies

$$\mathbb{E}_{\bar{x}(k)\sim\mu_\pi} [\mathcal{C}(k)] \leq \sigma^k \frac{\tilde{\alpha}_2}{\tilde{\alpha}_1} \mathbb{E}_{\bar{x}(0)\sim\mu_\pi} [\mathcal{C}(0)].$$

Furthermore, we can obtain the following equation

$$\mathbb{E}_{\bar{x}(k)\sim\mu_\pi} [x(k) - x_r(k)] \leq \sigma^k \frac{\tilde{\alpha}_2}{\tilde{\alpha}_1} \mathbb{E}_{\bar{x}(0)\sim\mu_\pi} [x(0) - x_r(0)].$$

Since we have designed a controller for the reference model, which can drive $x_r(k) = 0$, the following inequality can be obtained

$$\mathbb{E}_{\bar{x}(k)\sim\mu_\pi} [x(k)] \leq \sigma^k \frac{\tilde{\alpha}_2}{\tilde{\alpha}_1} \mathbb{E}_{\bar{x}(0)\sim\mu_\pi} [x(0)].$$

Therefore, the system states using the learned policy exponentially converge. The proof is completed. ■

In Theorem 2, the stability of the system (3) is proved. Since a data-based design and analysis approach is used in this paper, how to deal with attacks in the proof is implicit, which is different from existing results, such as Jin et al. (2017), Yucelen et al. (2016). The authors of Jin et al. (2017), Yucelen et al. (2016) proposed an adaptive scheme to suppress attacks and recover the system performance in a model-based manner. In our paper, a learned policy is used to deal with attacks and a constraint of the Lyapunov function is learned in our designed algorithm, by using which the proof of stability is completed in Theorem 2.

To improve the system performance, the passive performance index can be added to the constraint $\Delta\mathcal{L}(k)$ in (14). Based on such a constraint, system (3) can be proved to be passive, as described in Definition 2.

**Definition 2.** The system (3) is said to be passive if there exists a scalar $\rho > 0$ such that the following inequality holds

$$2\mathbb{E}\left[\sum_{k=0}^{T} y^T(k) \Gamma u_a(k)\right] \geq -\rho \sum_{k=0}^{T} (\Gamma u_a(k))^T (\Gamma u_a(k)),$$

where $y(k)$ means the output measurement.

The following corollary is given to show that system (3) can be passive if a policy can be learned under the constraint.

**Corollary 1.** *Given a constant $\rho > 0$, the system (3) is said to be passive if a policy can be learned and the following constraint holds by using Algorithm 1*

$$\begin{aligned}
&\mathbb{E}_{\bar{x}(k)\sim\mu_\pi} \left[ \mathbb{E}_{\bar{x}(k+1)\sim\mathcal{P}_\pi} [\mathcal{L}(k+1)] - \mathcal{L}(k) \right] \\
&\leq -\beta \mathbb{E}_{\bar{x}(k)\sim\mu_\pi} [\mathcal{C}(k)] \\
&\quad + \mathbb{E}_{\bar{x}(k)\sim\mu_\pi} \left[ 2y^T(k) \Gamma u_a(k) + \rho (\Gamma u_a(k))^T (\Gamma u_a(k)) \right].
\end{aligned}$$

**Proof.** The proof can refer to that in Theorem 2, which is omitted for want of space here. ■

## 5. Simulation

In this section, the simulation results are given to show the effectiveness and advantages of the proposed data-based secure control scheme. For the simulation, we use Python 3.6 and Tensorflow 1.15 to realize Algorithm 1. The details are given as follows.

**Example 1.** We assume the physical plant in Fig. 1 is a satellite control system, whose dynamics are described as Heydari and Balakrishnan (2012)

$$\dot{\omega} = J^{-1} \left( (u + d) - \omega \times J\omega \right),$$

where $\omega = [\omega_x \ \omega_y \ \omega_z]^T$ means the angular velocity vector, $J$ is the inertia tensor, $u$ is the control, and $d$ is the disturbance. The symbol "$\times$" means operation of the cross product.

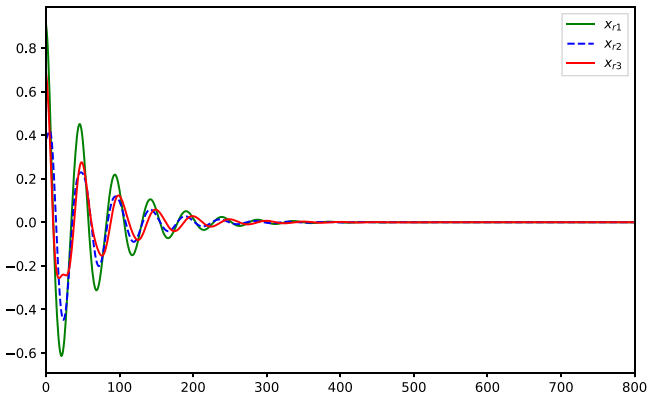


**Fig. 4.** Euler angle responses of the reference model using a PD controller.

The kinematic equation of the satellite can be described as

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta)\\ 0 & \cos(\phi) & -\sin(\phi)\\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)}\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix},$$

where $\phi$, $\theta$ and $\psi$ are the Euler angles of the satellite attitude, which correspond to $x$, $y$ and $z$ axes of the inertial coordinate system.

Then, the state equation of the Euler angles and the angular velocities is described as

$$\dot{x} = f(x,u)$$

where

$$x = \begin{bmatrix}\phi & \theta & \psi & \omega_x & \omega_y & \omega_z\end{bmatrix}^T,$$
$$f(x,u) = \begin{bmatrix}\mathscr{A}\\ -J^{-1}(\omega \times J\omega)\end{bmatrix} + \begin{bmatrix}0\\ J^{-1}\end{bmatrix}u,$$
$$\mathscr{A} = \begin{bmatrix}1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta)\\ 0 & \cos(\phi) & -\sin(\phi)\\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)}\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix}.$$

We assume that an adversary hijacks into the cyber layer, and injects false data into the control signal to deteriorate the satellite attitude. The control objective is to guarantee the stability of the satellite under attacks. For the above state equation, the sampling period is given as $\Delta t = 0.2$ s.

For the reference model, it can be obtained by setting the inertia tensor $J = \text{diag}\{108, 101, 114\}$ kg m$^2$. A PD controller is designed for the reference model in the following form

$$u_r(k) = -K_p\begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^T - K_d\begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^T,$$

where $K_p = 45$ and $K_d = 25$.

Fig. 4 depicts the Euler angle responses of the reference model using the PD controller. In the following, we will first give the setup of the deep neural networks to be trained. Neural networks will be trained by using Algorithm 1. Then, we show that the control signal sampled from the learned policy can stabilize the Euler angle responses without attacks, the performance of which can match that shown in Fig. 4.

Before setting the training parameters, the inertial tensor $J$ practically used in the system is defined as follows

$$J = \begin{bmatrix}100 & 2 & 0.5\\ 2 & 100 & 1\\ 0.5 & 1 & 110\end{bmatrix}\ \text{kg m}^2$$

**Table 1**
Hyperparameters for Example 1.

| Hyperparameters | Value |
|---|---|
| Length of sampling trajectories | 800 |
| Minibatch size | 256 |
| Actor learning rate | 1e−4 |
| Critic learning rate | 3e−4 |
| $\alpha$ learning rate | 1e−4 |
| $\lambda$ learning rate | 3e−4 |
| Target entropy | −3 |
| Soft replacement($\tau$) | 0.005 |
| Discount factor ($\gamma$) | 0.999 |
| $\beta$ | 0.1 |
| Structure of deep neural networks for actor | (128, 64, 32) |
| Structure of deep neural networks for critic | (256, 128, 64) |

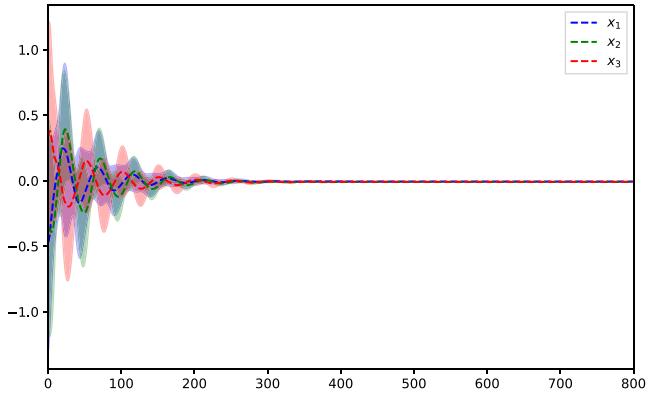


**Fig. 5.** Mean responses of Euler angles of the satellite system using the learned controller under different initial conditions.

In the training process, we train 10 policies, from which we choose the best one to stabilize the satellite attitude. The parameters used in the training are given in Table 1. The initial condition for the Euler angles is randomly sampled from [−1.5 1.5] rad, and the initial condition for angular velocities is randomly sampled from [−0.2 0.2] rad/s. Additionally, considering that the aim of attackers is to maximize the attack performance, an optimal attack scenario is considered in the training.

After training, we choose the best policy from the trained results. Then, 10 inferences are executed by using the chosen policy without considering attacks. Fig. 5 provides the mean responses of Euler angles under 10 different initial conditions using the learned controller. Fig. 6 provides the mean responses of Euler angles under 10 different initial conditions using the PD controller. As can be seen from Fig. 5, the satellite attitude without attacks can be well stabilized using the proposed learning algorithm.

As discussed in previous sections, there exist three categories of false data injection attacks, that is, constant attacks (reset attacks) (Ni, Guo, Mo, & Shi, 2019), time-dependent attacks (Jin et al., 2017), and optimal attacks (Guo et al., 2016; Wu, Sun and Chen, 2018). Next, these three attack scenarios are given to show the effectiveness of the proposed scheme. Also, to show the advantages of the proposed control scheme, comparisons with a PD controller are provided. In the following simulation figures, the shadowed region can be regarded as a confidence interval, which is obtained by calculating the mean and standard deviation of the trajectories.

*Case 1* Random constant attacks

In this case, we assume that an adversary randomly samples attack signals from [−5, 5]. Attacks are implemented from the initial time constant. Similar to the above simulation results, 10 inferences are executed under different initial conditions. Figs. 7

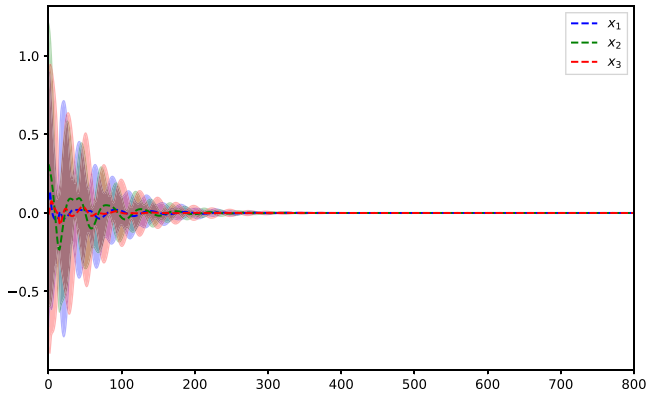


**Fig. 6.** Mean responses of Euler angles of the satellite system using a PD controller under different initial conditions.

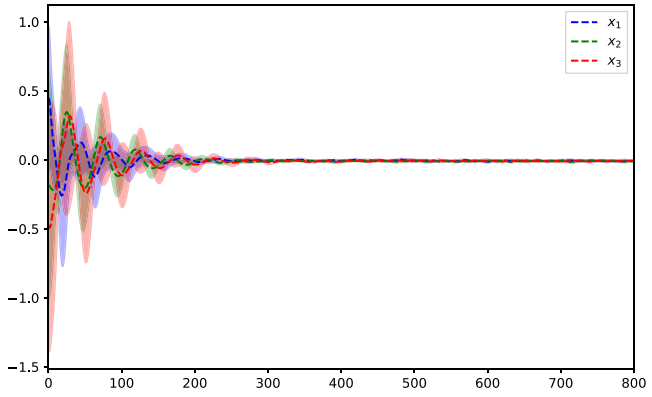


**Fig. 7.** Mean responses of Euler angles of the satellite system under random attacks using the learned controller under different initial conditions.

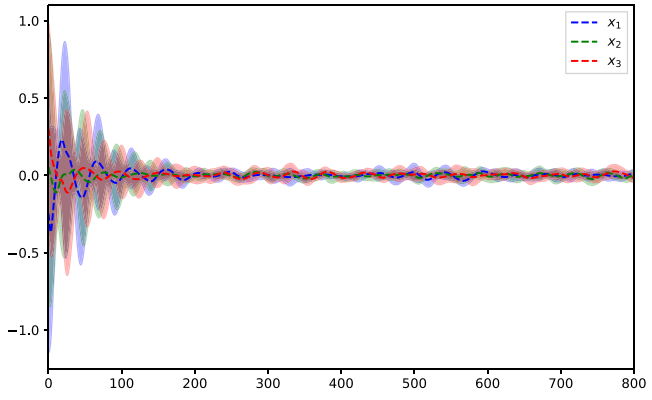


**Fig. 8.** Mean responses of Euler angles of the satellite system under attacks using a PD controller under different initial conditions.

and 8 respectively depict the mean responses of Euler angles using the learned policy and a PD controller. As can be seen from these two figures, the proposed learning algorithm can effectively mitigate attacks.

*Case 2* Time-dependent attacks

In this situation, we assume an adversary fulfill time-dependent attacks for $k \geq 0$, and

$$u_a(k) = \begin{bmatrix} \cos(0.1k) & 0.3\sin(k) - 1.5 & 2 + \cos(k) \end{bmatrix}^T.$$

Figs. 9 provides the mean responses of Euler angles of the satellite under the learned controller. The mean responses of

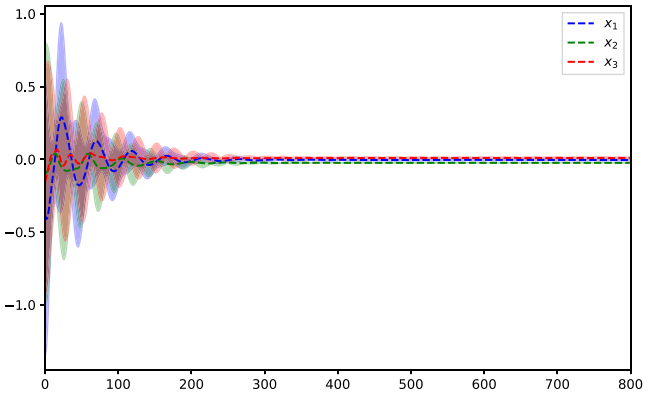


**Fig. 9.** Mean responses of Euler angles of the satellite system under attacks using a learned controller under different initial conditions.

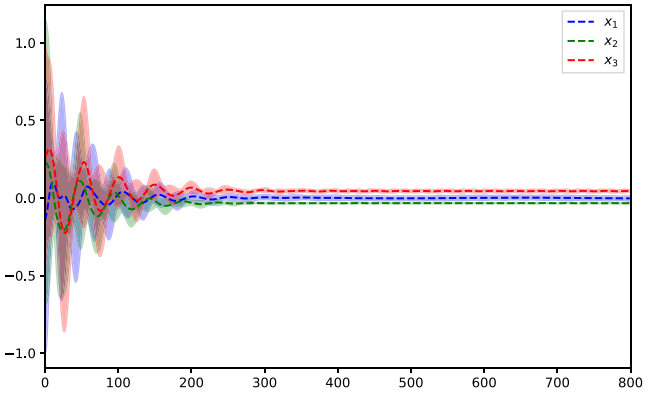


**Fig. 10.** Mean responses of Euler angles of the satellite system under attacks using a PD controller under different initial conditions.

Euler angles of the satellite under the PD controller are given in 10. As can be seen from Fig. 10, the time-dependent attacks change the attitude of the satellite. Although a robust PD controller is deployed, the attitude of the satellite is not recovered. Comparing these two figures, we can know that the proposed controller in this paper effectively mitigates the attacks.

*Case 3.* Optimal attacks

For the adversary, it utilizes the system knowledge to design attacks. The optimal attacks are constructed as

$$u_a(k) = K_a \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T,$$

where $K_a$ is the attack distribution matrix, and

$$K_a = \begin{bmatrix} -6 & 12 & -2 \\ -14 & -28 & 10 \\ -8 & 16 & -2 \end{bmatrix}.$$

The simulation results under optimal attacks are provided in Figs. 11 and 12, in which Fig. 11 depicts the mean trajectories of Euler angles using a learned controller, and mean responses of Euler angles using a PD controller are shown in Fig. 12. Apparently, the attitude of satellite is severer deteriorated when the PD controller is used yet our learned controller can mitigate attacks and maintain the desired performance. Also, as can be seen from Figs. 7, 9 and 11, the learned secure controller can achieve the best performance in *Case 3*, which may depend on the setting of attack pattern in the training.

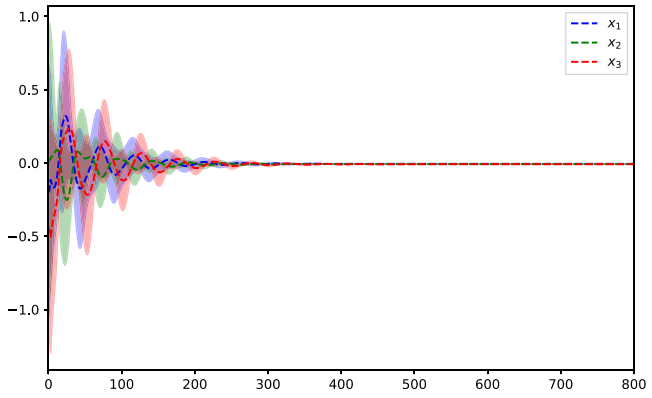


**Fig. 11.** Mean responses of Euler angles of the satellite system under attacks using a learned controller under different initial conditions.

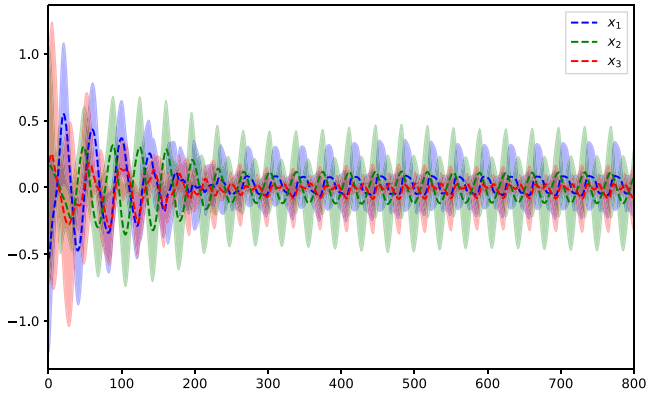


**Fig. 12.** Mean responses of Euler angles of the satellite system under attacks using a PD controller under different initial conditions.

From the above rich simulation results, we can conclude that the proposed learning algorithm outperforms a well-designed PD controller. Relying on the superiorities of deep neural networks, the learning-based control scheme can deal with different categories of false data injection attacks. It is noted that the proposed scheme can fail to deal with the situation, which it is not experienced. If generative adversarial networks can be introduced, the performance of the proposed algorithm can be greatly improved.

Next, a mechanical system is utilized to validate the effectiveness of Corollary 1.

**Example 2.** Considering the physical plant in Fig. 1 is a single-link robot arm, whose dynamics are governed by the following equation (Cao, Niu, & Song, 2019)

$$\ddot{\theta} = -\frac{gLM}{J}\sin(\theta) - \frac{D(t)}{J}\dot{\theta} + \frac{1}{J}u(t),$$

where $\theta$ means the angle position, $\dot{\theta}$ means the angle velocity, $g = 9.8$ m/s$^2$ is the acceleration of gravity, $L = 0.5$ m means the length of the arm, $M = 10$ kg denotes the mass of payload, $J = 10$ kg m$^2$ is the moment of inertia, $D(t) = 2 + 0.2\sin(t)$ means the uncertain but bounded coefficient of viscous friction, and $u(t)$ is the control input.

Define $x = [x_1\ x_2]^T = [\theta\ \dot{\theta}]^T$, and the above equation can be described as

$$\dot{x} = \begin{bmatrix} 0 & x_2 \\ \frac{-gLM}{J}\sin(x_1) & \frac{-D(t)}{J}x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} u(t).$$

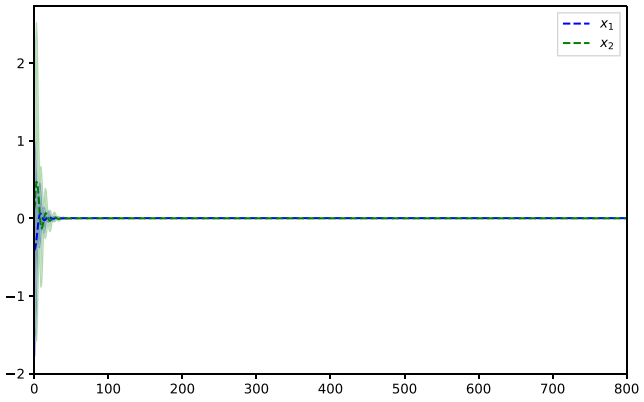


**Fig. 13.** Mean responses of states without attacks using a learned controller under different initial conditions.

**Table 2**
Hyperparameters for Example 2.

| Hyperparameters | Value |
|---|---|
| Length of sampling trajectories | 800 |
| Minibatch size | 256 |
| Actor learning rate | 2e−4 |
| Critic learning rate | 5e−4 |
| $\alpha$ learning rate | 2e−4 |
| $\lambda$ learning rate | 5e−4 |
| Target entropy | −1 |
| Soft replacement($\tau$) | 0.005 |
| Discount factor ($\gamma$) | 0.999 |
| $\beta$ | 0.1 |
| Structure of deep neural networks for actor | (64, 32) |
| Structure of deep neural networks for critic | (128, 64) |

For the reference model, it can be obtained by linearizing the above nonlinear system (Cao et al., 2019). Then, a linear quadratic optimal controller can be readily designed. An adversary intends to inject false data into the control signal to deteriorate the system performance. Next, a secure policy is trained to mitigate the attacks.

Table 2 provides the hyperparameters used in the training. In the training, we also learn 10 policies, from which we choose the best one as our policy to stabilize the robot arm under attacks. The initial condition for $x_1$ satisfies a uniform distribution $[-2\ 2]$ rad, and $x_2$ satisfies a uniform distribution $[-1\ 1]$ rad/s. The sampling period is defined as 0.2 s.

In this example, a PD controller for the robot arm is designed as $u_{PD}(k) = -5x_1(k) - 20x_2(k)$, and we choose the best policy from 10 learned policies. First, the effectiveness of the designed PD controller and the learned controller is evaluated without considering attacks. Using the learned controller, Fig. 13 shows the mean responses of the robot arm states without attacks under 10 under random initial conditions. The mean responses under the designed PD controller are provided in Fig. 14. As can be seen from these simulation results, we can conclude that the learned controller can effectively stabilize the robot arm.

Next, the attack situation is addressed to show the effectiveness and advantages of the learned controller. We assume that the adversary constructs the attack signal as $u_a(k) = -5\sin(x_1(k)) + 10x_2$. Under such attack signals, Fig. 15 gives the mean responses of states using the learned controller, and the mean responses of states using the designed PD controller are provided in Fig. 16, which shows that the designed PD controller is not capable of dealing with such attacks. By comparing the simulation results in these two figures, we can conclude that the learned controller can mitigate such attacks effectively.

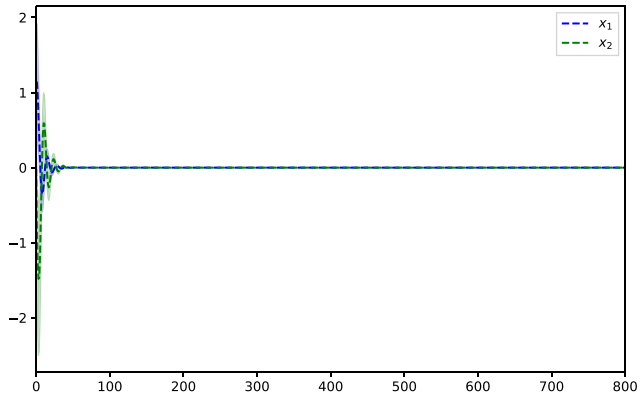


**Fig. 14.** Mean responses of states without attacks using a PD controller under different initial conditions.

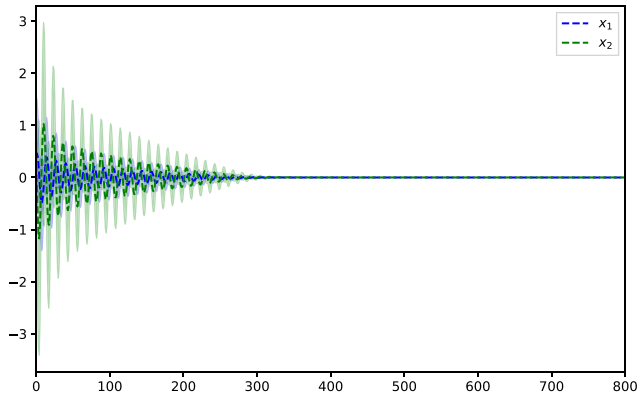


**Fig. 15.** Mean responses of states under attacks using a learned controller under different initial conditions.

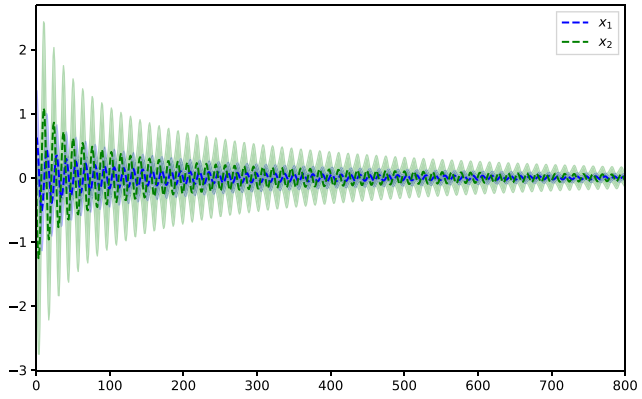


**Fig. 16.** Mean responses of states under attacks using a PD controller under different initial conditions.

## 6. Conclusion

This paper has developed a novel data-based secure control scheme for CPS under false data injection attacks. The secure control problem has been formulated as an MDP, which has been solved by proposing a Lyapunov-based soft actor–critic deep reinforcement learning algorithm. Deep neural networks constructed by fully connected multiple layer perceptrons have been used to approximate the "actor" and the "critic". By defining a Lyapunov candidate as the "critic", a constraint on the Lyapunov function has been added to the training process. By using the learned Lyapunov function, the stability of CPS under attacks has been proven. Simulation results have been provided to show the effectiveness and advantages of the proposed secure control scheme in this paper.

## References


Abbaspour, A., Sargolzaei, A., Forouzannezhad, P., Yen, K. K., & Sarwat, A. I. (2020). Resilient control design for load frequency control system under false data injection attacks. *IEEE Transactions on Industrial Electronics*, *67*, 7951–7962.

Abrams, M., & Weiss, J. (2008). *Malicious control system cyber security attack case study–maroochy water services, Australia*. McLean, VA: The MITRE Corporation.

Cao, Z., Niu, Y., & Song, J. (2019). Finite-time sliding-mode control of Markovian jump cyber–physical systems against randomly occurring injection attacks. *IEEE Transactions on Automatic Control*, *65*, 1264–1271.

Cardenas, A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., Sastry, S., et al. (2009). Challenges for securing cyber physical systems. In *Workshop on future directions in cyber–physical systems security, Vol. 5*. Citeseer.

Chen, J., Touati, C., & Zhu, Q. (2019). A dynamic game approach to strategic design of secure and resilient infrastructure network. *IEEE Transactions on Information Forensics and Security*, *15*, 462–474.

Chow, Y., Nachum, O., Duenez-Guzman, E., & Ghavamzadeh, M. (2018). A lyapunov-based approach to safe reinforcement learning. In *Advances in neural information processing systems* (pp. 8092–8101).

Dahl, G. E., Sainath, T. N., & Hinton, G. E. (2013). Improving deep neural networks for LVCSR using rectified linear units and dropout. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8609–8613). IEEE.

De Persis, C., & Tesi, P. (2015). Input-to-state stabilizing control under denial-of-service. *IEEE Transactions on Automatic Control*, *60*, 2930–2944.

Ding, D., Han, Q.-L., Wang, Z., & Ge, X. (2019). A survey on model-based distributed control and filtering for industrial cyber–physical systems. *IEEE Transactions on Industrial Informatics*, *15*, 2483–2499.

Dolk, V., Tesi, P., De Persis, C., & Heemels, W. (2016). Event-triggered control systems under denial-of-service attacks. *IEEE Transactions on Control of Network Systems*, *4*, 93–105.

Farwell, J. P., & Rohozinski, R. (2011). Stuxnet and the future of cyber war. *Survival*, *53*, 23–40.

Fawzi, H., Tabuada, P., & Diggavi, S. (2014). Secure estimation and control for cyber–physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, *59*, 1454–1467.

Feng, Z., & Hu, G. (2019). Secure cooperative event-triggered control of linear multiagent systems under dos attacks. *IEEE Transactions on Control Systems Technology*, *28*, 741–752.

Giraldo, J., Urbina, D., Cardenas, A., Valente, J., Faisal, M., Ruths, J., et al. (2018). A survey of physics-based attack detection in cyber–physical systems. *ACM Computing Surveys*, *51*, 1–36.

Guo, Z., Shi, D., Johansson, K. H., & Shi, L. (2016). Optimal linear cyber-attack on remote state estimation. *IEEE Transactions on Control of Network Systems*, *4*, 4–13.

Gupta, S. K. (1995). *Numerical methods for engineers*. New Age International.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv preprint arXiv:1801.01290.

Han, M., Zhang, L., Wang, J., & Pan, W. (2020). Actor-critic reinforcement learning for control with stability guarantee. arXiv preprint arXiv:2004.14288.

Heydari, A., & Balakrishnan, S. N. (2012). Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics. *IEEE Transactions on Neural Networks and Learning Systems*, *24*, 145–157.

Hu, L., Wu, C., & Pan, W. (2020). Lyapunov-based reinforcement learning state estimator. arXiv preprint arXiv:2010.13529.

Jin, X., Haddad, W. M., & Yucelen, T. (2017). An adaptive control architecture for mitigating sensor and actuator attacks in cyber–physical systems. *IEEE Transactions on Automatic Control*, *62*, 6058–6064.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Lee, E. A. (2008). Cyber physical systems: Design challenges. In *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)* (pp. 363–369). IEEE.

Li, X.-M., Zhou, Q., Li, P., Li, H., & Lu, R. (2019). Event-triggered consensus control for multi-agent systems against false data-injection attacks. *IEEE Transactions on Cybernetics*, *50*, 1856–1866.

Liang, G., Weller, S. R., Zhao, J., Luo, F., & Dong, Z. Y. (2016). The 2015 Ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems*, *32*, 3317–3318.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.
Lu, A.-Y., & Yang, G.-H. (2019). Switched projected gradient descent algorithms for secure state estimation under sparse sensor attacks. *Automatica*, *103*, 503–514.
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*, 529–533.
Mo, Y., & Sinopoli, B. (2015). On the performance degradation of cyber–physical systems under stealthy integrity attacks. *IEEE Transactions on Automatic Control*, *61*, 2618–2624.
Ni, Y., Guo, Z., Mo, Y., & Shi, L. (2019). On the performance analysis of reset attack in cyber–physical systems. *IEEE Transactions on Automatic Control*, *65*, 419–425.
Perkins, T. J., & Barto, A. G. (2001). Lyapunov-constrained action sets for reinforcement learning. In *ICML, Vol. 1* (pp. 409–416).
Perkins, T. J., & Barto, A. G. (2002). Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, *3*, 803–832.
Qin, J., Li, M., Shi, L., & Yu, X. (2018). Optimal denial-of-service attack scheduling with energy constraint over packet-dropping networks. *IEEE Transactions on Automatic Control*, *63*, 1648–1663.
Royden, H. L. (1968). *Real analysis*. Krishna Prakashan Media.
Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning* (pp. 1889–1897).
Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: an introduction*. MIT Press.
Teixeira, A. M. (2019). Optimal stealthy attacks on actuators for strictly proper systems. In *2019 IEEE 58th conference on decision and control (CDC)* (pp. 4385–4390). IEEE.
Teixeira, A., Pérez, D., Sandberg, H., & Johansson, K. H. (2012). Attack models and scenarios for networked control systems. In *Proceedings of the 1st international conference on high confidence networked systems* (pp. 55–64).
Wu, C., Hu, Z., Liu, J., & Wu, L. (2018). Secure estimation for cyber–physical systems via sliding mode. *IEEE Transactions on Cybernetics*, *48*, 3420–3431.
Wu, C., Li, X., Pan, W., Liu, J., & Wu, L. (2021). Zero-sum game based optimal secure control under actuator attacks. *IEEE Transactions on Automatic Control*, *66*, 3773–3780.
Wu, G., Sun, J., & Chen, J. (2018). Optimal data injection attacks in cyber–physical systems. *IEEE Transactions on Cybernetics*, *48*, 3302–3312.
Wu, C., Wu, L., Liu, J., & Jiang, Z.-P. (2019). Active defense-based resilient sliding mode control under denial-of-service attacks. *IEEE Transactions on Information Forensics and Security*, *15*, 237–249.
Wu, C., Yao, W., Pan, W., Sun, G., Liu, J., & Wu, L. (2022). Secure control for cyber–physical systems under malicious attacks. *IEEE Transactions on Control of Network Systems*, *9*, 775–788.
Yucelen, T., Haddad, W. M., & Feron, E. M. (2016). Adaptive control architectures for mitigating sensor attacks in cyber–physical systems. *Cyber-Physical Systems*, *2*, 24–52.
Zhang, Q., Dong, H., & Pan, W. (2020). Lyapunov-based reinforcement learning for decentralized multi-agent control. arXiv preprint arXiv:2009.09361.
Zhang, Q., Pan, W., & Reppa, V. (2020). Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles. arXiv preprint arXiv:2008.07240.
Zhang, R., & Venkitasubramaniam, P. (2017). Stealthy control signal attacks in linear quadratic gaussian control systems: Detectability reward tradeoff. *IEEE Transactions on Information Forensics and Security*, *12*, 1555–1570.
Zhang, H., & Zheng, W. X. (2018). Denial-of-service power dispatch against linear quadratic control via a fading channel. *IEEE Transactions on Automatic Control*, *63*, 3032–3039.

**Chengwei Wu** received the Ph.D. degree from Harbin Institute of Technology, China, 2021. From July 2015 to December 2015, he was a Research Assistant in the Department of Mechanical Engineering, The Hong Kong Polytechnic University. From 2019 to 2021, he was a joint-PhD student at Department of Cognitive Robotics, Delft University of Technology, Netherlands. He is currently an Assistant Professor with the Harbin Institute of Technology, Harbin, China. His research interests include sliding mode control, deep reinforcement learning and cyber–physical systems.

**Wei Pan** received the Ph.D. degree in Bioengineering from Imperial College London, UK, 2017. He is currently a Senior Lecturer (Associate Professor) in Machine Learning at the Department of Computer Science and a member of Centre for AI Fundamentals and Centre for Robotics and AI, University of Manchester, UK. Before that, he was an Assistant Professor in Robot Dynamics at the Department of Cognitive Robotics and co-director of Delft SELF AI Lab, Delft University of Technology, Netherlands and a Project Leader at DJI, China. He is an Area Chair or Associate Editor of *IEEE Robotics and Automation Letters, ACM Transactions on Probabilistic Machine Learning, Conference on Robot Learning (CoRL), IEEE International Conference on Robotics and Automation (ICRA), IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. He has broad interest in robot control using machine learning and the principles of dynamic control.

**Rick Staa** received his B.Sc. degree in Human Movement Science from Vrije Universiteit Amsterdam in 2016. He is currently a Graduate student at the Department of Cognitive Robotics, Delft University of Technology. His Master thesis includes topics like reinforcement learning, robot dynamics and robot control.

**Jianxing Liu** received the B.S. degree in mechanical engineering in 2008, the M.E. degree in control science and engineering in 2010, both from Harbin Institute of Technology, Harbin, China and the Ph.D. degree in Automation from the Technical University of Belfort-Montbeliard (UTBM), France, in 2014. Since 2014, he joined Harbin Institute of Technology, Harbin, China. His current research interests include nonlinear control algorithms, sliding mode control, and their applications in industrial electronics systems and renewable energy systems.

**Guanghui Sun** received the B.S. degree in automation and the M.S. and Ph.D. degrees in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2005, 2007, and 2010, respectively. He is currently a Professor in the Department of Control Science and Engineering, Harbin Institute of Technology. His research interests include fractional-order systems, networked control systems, and sliding mode control.

**Ligang Wu** received the B.S. degree in Automation from Harbin University of Science and Technology, China in 2001; the M.E. degree in Navigation Guidance and Control from Harbin Institute of Technology, China in 2003; the Ph.D. degree in Control Theory and Control Engineering from Harbin Institute of Technology, China in 2006. From January 2006 to April 2007, he was a Research Associate in the Department of Mechanical Engineering, The University of Hong Kong, Hong Kong. From September 2007 to June 2008, he was a Senior Research Associate in the Department of Mathematics, City University of Hong Kong, Hong Kong. From December 2012 to December 2013, he was a Research Associate in the Department of Electrical and Electronic Engineering, Imperial College London, London, UK. In 2008, he joined the Harbin Institute of Technology, China, as an Associate Professor, and was then promoted to a Full Professor in 2012. Prof. Wu was the winner of the National Science Fund for Distinguished Young Scholars in 2015, and received China Young Five Four Medal in 2016. He was named as the Distinguished Professor of Chang Jiang Scholar in 2017, and was named as the Highly Cited Researcher in 2015–2019.

Prof. Wu currently serves as an Associate Editor for a number of journals, including *IEEE/ASME Transactions on Mechatronics, IEEE Transactions on Industrial Electronics, IEEE Industrial Electronics Magazine, IEEE Transactions on Industrial Cyber-Physical Systems, Information Sciences, Signal Processing,* and *IET Control Theory and Applications* He is an Associate Editor for the Conference Editorial Board, IEEE Control Systems Society. He is also a Fellow of IEEE.

Prof. Wu has published 8 research monographs and more than 200 research papers in international referred journals. His current research interests include switched systems, stochastic systems, computational and intelligent systems, sliding mode control, and advanced control techniques for power electronic systems.