SINR-Based DoS Attack on Remote State Estimation: A Game-Theoretic Approach

Yuzhe Li, *Member, IEEE*, Daniel E. Quevedo, *Senior Member, IEEE*,
Subhrakanti Dey, *Senior Member, IEEE*, and Ling Shi, *Member, IEEE*

***Abstract*—We consider remote state estimation of cyberphysical systems under signal-to-interference-plus-noise ratio-based denial-of-service attacks. A sensor sends its local estimate to a remote estimator through a wireless network that may suffer interference from an attacker. Both the sensor and the attacker have energy constraints. We first study an associated two-player game when multiple power levels are available. Then, we build a Markov game framework to model the interactive decision-making process based on the current state and information collected from previous time steps. To solve the associated optimality (Bellman) equations, a modified Nash Q-learning algorithm is applied to obtain the optimal solutions. Numerical examples and simulations are provided to demonstrate our results.**

***Index Terms*—Cyberphysical systems, game theory, remote state estimation, security, wireless sensors.**

I. Introduction

Cyberphysical systems (CPS) have attracted considerable interest from both academic and industrial communities in the past few years. With the integration of sensing, control, communication, and computation, a wide application spectrum of CPSs are found, such as smart grid, intelligent transportation, and environmental monitoring [1]. Wireless sensors are key components in CPS and have advantages, such as low cost, easy installation, and self-power [2], [3], when compared with traditional wired sensors. However, due to the use of wireless networks, wireless sensors are more vulnerable to cybersecurity threats than wired sensors. In addition, the increasing penetration of CPS to safety-critical infrastructures of the society increases the risks and severities of such attacks. Therefore, the security issue is of fundamental importance to ensure the safe operation of CPS.

Two possible types of attacks on CPS are commonly investigated in the literature: deception (integrity) attacks and denial-of-service (DoS) attacks [4], corresponding to the two traditional security goals *integrity* and *availability*, respectively. Typically, the integrity attacks require comprehensive information about the system and modifications of the data. Such information is not needed for DoS attacks, making them a more reachable (and likely) alternative for the adversary. Different from most traditional computer systems where DoS attacks cannot cause serious damage, some critical systems in CPS, which rely on real-time operation, may become unstable and even be damaged under DoS attacks [4]. Due to the dynamic nature of CPS, when the attacker and defender choose actions, they should take consideration of the actions that their opponent may take. Therefore, instead of a static analysis focusing on only one side of the security issues [5], a more comprehensive game-theoretic framework to model the interactive action making process between both sides is needed [6], [7]. Previous approaches to studying DoS attacks in CPS using game theory can be found in [8]–[11].

In many existing works, such as [5], [9], [12], the DoS attack is modelled as a binary process considering sending or not for the sensor and blocking or not for the attacker. To elaborate on the interactive process, we extend the model to a SINR-based network where both the sensor and the attacker can choose their actions with multiple energy levels. The proposed problem formulation also addresses the power control issue for wireless sensors which are usually expected to work for a long time without replacements of the onboard batteries (e.g., due to widespread sensors and a possibly dangerous environment [1]). Therefore, the sensors face a tradeoff between consuming more energy to increase link reliability thereby ensuring accurate remote estimation performance, and consuming less energy to meet energy constraints. The attackers face a similar situation: they may also have limited resources but want to worsen the system performance as much as possible. In this paper, we aim to study the transmission power strategy for the sensor and the interference power strategy for the attacker, and their equilibrium under a game-theoretic framework. The main contributions of this paper are summarized as follows.

1) **SINR-based Power Control**: We study the interactions between the transmission power of the sensor and the interference power of the DoS attacker through an SINR-based network model.
2) **Estimation Quality-based Objectives**: In our work, we study the behaviors of the sensor and the attacker with an integrated objective combining the communication cost and estimation error covariance, rather than studying these two important components separately.

Manuscript received February 2, 2015; revised September 12, 2015 and January 15, 2016; accepted March 21, 2016. Date of publication April 1, 2016; date of current version September 15, 2017. The work by Y. Li and L. Shi is supported by a HK RGC theme-based project T23-701/14N. Recommended by Associate Editor M. Mesbahi.

Y. Li and L. Shi are with the Electronic and Computer Engineering Department, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: yliah@ust.hk; eesling@ust.hk).

D. E. Quevedo is with the Department of Electrical Engineering (EIM-E), Paderborn University, 33100 Paderborn, Germany.

S. Dey is with the Department of Engineering Sciences, Uppsala University, 75121 Sweden (e-mail: subhrakanti.dey@angstrom.uu.se).

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/TCNS.2016.2549640

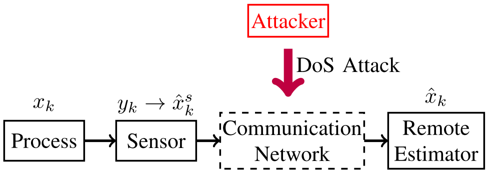


Fig. 1. Communication network is jammed by a malicious attacker.

3) **Markov Game Framework**: In addition to studying an extension of the problem in [13], we also consider the scenario where both sides make online interactive decisions through a Markov game framework. To solve the associated optimality (Bellman) equations, a modified Nash Q-learning algorithm is proposed and applied.

The remainder of this paper is organized as follows. Section II presents the system framework and states the main problem of interest. Section III considers an extension of the problem in [13] where multiple power levels are available. Section IV sets up the framework for the Markov game and provides a modified Nash Q-learning algorithm to obtain the optimal solutions. Numerical examples and simulations are demonstrated in Section V. Section VI provides some concluding remarks.

Notations: $\mathbb{Z}$ denotes the set of all integers and $\mathbb{N}$ the positive integers. $\mathbb{R}$ is the set of real numbers. $\mathbb{R}^n$ is the n-dimensional Euclidean space. $\mathbb{S}_+^n$ (and $\mathbb{S}_{++}^n$) is the set of n by n positive semidefinite matrices (and positive definite matrices). When $X \in \mathbb{S}_+^n$ (and $\mathbb{S}_{++}^n$), we write $X \geqslant 0$ (and $X > 0$). $X \geqslant Y$ if $X - Y \in \mathbb{S}_+^n$. The curled inequality symbols $\succeq$ and $\preceq$ (and their strict forms $\succ$ and $\prec$) are used to denote generalized componentwise inequalities between vectors: for vectors $\mathbf{a} = [a_1, a_2, \ldots, a_n]'$, $\mathbf{b} = [b_1, b_2, \ldots, b_n]'$, we write $\mathbf{a} \succeq \mathbf{b}$ if $a_i \geqslant b_i$, for $i = 1, 2, \ldots, n$. $\mathbf{1}$ denotes a vector with all entries one. $\mathrm{Tr}(\cdot)$ is the trace of a matrix. The superscript $'$ stands for transposition. For functions g, h with appropriate domains, $g \circ h(x)$ stands for the function composition $g(h(x))$, and $h^n(x) \triangleq h(h^{n-1}(x))$, where $n \in \mathbb{N}$ and with $h^0(x) \triangleq x$. δ_{ij} is the Dirac delta function, that is, δ_{ij} is equal to 1 when $i = j$, and 0 otherwise. The notation $\mathbb{P}[\cdot]$ refers to probability and $\mathbb{E}[\cdot]$ to expectation.

II. Problem Setup

Consider a general discrete-time linear time-invariant (LTI) process of the form

$$x_{k+1} = Ax_k + w_k, \quad y_k = Cx_k + v_k \tag{1}$$

where $k \in \mathbb{N}$, $x_k \in \mathbb{R}^{n_x}$ is the process state vector at time k, $y_k \in \mathbb{R}^{n_y}$ is the measurement taken by the sensor, $w_k \in \mathbb{R}^{n_x}$ and $v_k \in \mathbb{R}^{n_y}$ are zero-mean i.i.d. Gaussian noises with $\mathbb{E}[w_k w_j'] = \delta_{kj} Q (Q \geqslant 0)$, $\mathbb{E}[v_k v_j'] = \delta_{kj} R (R > 0)$, $\mathbb{E}[w_k v_j'] = 0\ \forall j, k \in \mathbb{N}$. The initial state x_0 is a zero-mean Gaussian random vector uncorrelated with w_k and v_k with covariance $\Pi_0 \geqslant 0$. The pair (A, C) is assumed to be observable and $(A, Q^{1/2})$ is controllable.

A. Local State Estimation

Our interest lies in the security of remote state estimation as depicted in Fig. 1. We consider the so-called "smart sensor" as described in [14], which first locally estimates the state x_k based on all measurements it has collected up to time k and then transmits its local estimate to the remote estimator.

Denote $\hat{x}_k^s$ and P_k^s as the sensor's local minimum mean-squared error (MMSE) estimate of the state x_k and the corresponding error covariance

$$\hat{x}_k^s = \mathbb{E}[x_k | y_1, y_2, \ldots, y_k] \tag{2}$$

$$\hat{P}_k^s = \mathbb{E}\left[(x_k - \hat{x}_k^s)(x_k - \hat{x}_k^s)' | y_1, y_2, \ldots, y_k\right] \tag{3}$$

which can be calculated by a standard Kalman filter.

Define the Lyapunov and Riccati operators $h, \tilde{g} : \mathbb{S}_+^n \to \mathbb{S}_+^n$ for notational ease as $h(X) \triangleq AXA' + Q$, $\tilde{g}(X) \triangleq X - XC'[CXC' + R]^{-1}CX$.

As the estimation error covariance of the Kalman filter converges to a unique value from any initial condition ([15]), without loss of generality, we assume that the Kalman filter at the sensor side has entered the steady state and we simplify our subsequent discussion by setting

$$P_k^s = \overline{P}, \quad k \geqslant 1 \tag{4}$$

where $\overline{P}$ is the steady-state error covariance given by the unique positive semidefinite solution of $\tilde{g} \circ h(X) = X$ (see [15]).

B. Communication Model With SINR

After obtaining $\hat{x}_k^s$, the sensor will transmit it as a data packet to the remote estimator. Due to fading and interference, random data drops will occur. To model this situation, assume that the communication between the sensor and the remote estimator is over an Additive White Gaussian Noise (AWGN) network using Quadrature Amplitude Modulation (QAM). Then the relationship between the symbol error rate (SER) and signal to noise ratio (SNR) is revealed by digital communication theory as the following (similar to [16], [17]):

$$\mathrm{SER} = 2Q(\sqrt{\alpha \mathrm{SNR}}), \quad Q(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-\eta^2/2) d\eta \tag{5}$$

and $\alpha > 0$ is a parameter.

In the presence of a DoS interference attacker, the SNR for the communication network [18] can be rewritten as $\mathrm{SINR} = (p_k/(\omega_k + \sigma^2))$, where p_k is the transmission power used by the sensor at time step k, σ^2 is the additive white Gaussian noise power, and ω_k is the interference power from the attacker.

We shall assume that the remote estimator can detect symbol errors via cyclic redundancy check (CRC) (see [18]). Thus, the transmission of $\hat{x}_k^s$ between the sensor and the remote estimator can be characterized by a binary random process $\{\gamma_k\}, k \in \mathbb{N}$

$$\gamma_k = \begin{cases} 1, & \text{if } \hat{x}_k^s \text{ arrives error-free at time } k \\ 0, & \text{otherwise (regarded as dropout).} \end{cases} \tag{6}$$

Based on (5) and the preceding discussion, we have

$$\lambda_k \triangleq \mathbb{P}[\gamma_k = 1] = 1 - 2Q\left(\sqrt{\alpha \frac{p_k}{\omega_k + \sigma^2}}\right). \tag{7}$$

Note that the SINR not only depends on the transmission power used by the sensor, but is also affected by the interference power from the DoS attacker. Different SINRs lead to different dropout rates and remote estimation performance as we shall see next.

C. Remote State Estimation

Denote $\hat{x}_k$ and P_k as the remote estimator's own MMSE state estimate and the corresponding error covariance based on all sensor data packets received up to time step k. They can be calculated based on the results in [16] and [19] via the following procedure: once the sensor's local estimate arrives, the estimator synchronizes $\hat{x}_k$ with that of the sensor, that is, with $\hat{x}_k^s$; otherwise, the remote estimator just predicts $\hat{x}_k$ based on its previous estimate using the system model (1). The remote state estimate $\hat{x}_k$ thus obeys the recursion

$$\hat{x}_k = \begin{cases} \hat{x}_k^s, & \text{if } \gamma_k = 1 \\ A\hat{x}_{k-1}, & \text{if } \gamma_k = 0. \end{cases} \tag{8}$$

The corresponding state estimation error covariance P_k satisfies

$$P_k = \begin{cases} \overline{P}, & \text{if } \gamma_k = 1 \\ h(P_{k-1}), & \text{if } \gamma_k = 0. \end{cases} \tag{9}$$

Note that due to the recursion of the dynamics in (9), P_k can only take value in the infinitely countable set $\{\overline{P}, h(\overline{P}), h^2(\overline{P}), \ldots\}$. We assume that the remote estimator will send ACKs to the sensor to indicate whether it has received the data packet successfully or not [17] at time k. This enables the sensor to compute P_{k-1} using (9).

The objective of the sensor is to design the transmission strategy to help the remote estimator obtain accurate state estimation. On the contrary, the attacker aims to deteriorate the remote estimation performance. When both sides have no access to the information of the real-time process, they can only design their strategies offline before the process starts, that is, independent of the state of the process. We will investigate this situation in Section III using a Markov chain model. On the other hand, when the online information is available, both the sensor and the attacker will make decisions based on the current state and information collected from previous time steps. This motivates us to consider situations where both sides make interactive decisions through a two-player Markov decision process in Section IV.

III. Finite Time-Horizon: Offline Game Framework

In this section, we consider a scenario similar to our previous work [13], where the sensor and the attacker design their strategies offline before the process starts, that is, independent of the state of the process. Different from [13] where the DoS attack is modelled as a binary process considering sending or not for the sensor and blocking or not for the attacker, we elaborate the interactive process by extending the model to a SINR-based network where both the sensor and the attacker can choose their actions with multiple energy levels.

A. Energy Constraints

We assume that both players are subject to the following energy constraints (the case without energy constraints is trivial as they will simply use the maximum power all the time):

$$\sum_{k=1}^{T} p_k = \mathcal{P} \tag{10}$$

$$\sum_{k=1}^{T} \omega_k = \mathcal{W} \tag{11}$$

where $\mathcal{P}, \mathcal{W} \in \mathbb{R}^+ \cup 0$. Note that the strategy sets for both sides are continuous and closed.

For convenience, denote the strategies of the sensor and attacker, respectively, as

$$\mathbf{p} = \{p_1, p_2, \ldots, p_T\} \tag{12}$$

$$\mathbf{w} = \{\omega_1, \omega_2, \ldots, \omega_T\}. \tag{13}$$

To analyze the optimal actions that the sensor and attacker can take under energy constraints, we first derive the expression for the objective function for both sides, which needs the following Markov chain model.

B. Markov Chain Model

The following definition is similar to [13].

Definition 3.1: Within the time-horizon T, if at time k, the state error covariance at the remote estimator $P_k = h^{i-1}(\overline{P})$, for some $i = 1, 2, \ldots, T+1$, then the state of the remote estimator is denoted as $S_k \triangleq z_{i,k}$. The state set for time k is defined as $\mathbf{Z}_k = \{S_k | S_k = z_{i,k}, 1 \leqslant i \leqslant T+1\}$, $k = 1, 2, \ldots, T$. Assume that $P_0 = \overline{P}$, that is, $\mathbf{Z}_0 = \{z_{1,0}\}$ is the initialization state set before the process begins. □

The transition matrix of the Markov chain $\{S_k\}$, $k = 1, 2, \ldots, T$ from state set $\mathbf{Z}_{k-1}$ to $\mathbf{Z}_k$ then can be defined as

$$\mathbf{T}_k(i_1, i_2) = \mathbb{P}\left[z_{i_2,k} | z_{i_1,k-1}\right]. \tag{14}$$

If the sensor data packet arrives at the remote estimator, that is, if $\gamma_k = 1$, then we have $P_k = \overline{P}$. Based on (14) and (7), this gives $\mathbf{T}_k(i_1, 1) = \mathbb{P}[z_{1,k} | z_{i_1,k-1}] = \mathbb{P}[P_k = \overline{P} | z_{i_1,k-1}] = \mathbb{P}[\gamma_k = 1] = \lambda_k$.

On the other hand, if the packet is dropped, then $P_k = h(P_{k-1})$, and we have $\mathbf{T}_k(i_1, i_1+1) = \mathbb{P}[z_{i_1+1,k} | z_{i_1,k-1}] = \mathbb{P}[P_k = h(P_{k-1}) | z_{i_1,k-1}] = \mathbb{P}[\gamma_k = 0] = 1 - \lambda_k$.

Other entries of $\mathbf{T}_k$ are 0, since the corresponding state transitions are not possible. This gives

$$\mathbf{T}_k = \begin{bmatrix} \lambda_k & 1-\lambda_k & & & \\ \lambda_k & & 1-\lambda_k & & \\ \vdots & & & \ddots & \\ \lambda_k & & & & 1-\lambda_k \end{bmatrix}_{(T+1)\times(T+1)}$$

where the missing entries are 0.

Define $\pi_{i,k}$ as the probability of state $z_{i,k}$ occurring at time k, that is

$$\pi_{i,k} = \mathbb{P}[S_k = z_{i,k}] \tag{15}$$

then we can construct the probability matrix $\Pi = [\pi_{i,k}]_{(T+1)\times T}$ based on the derivation in [13].

Once we have the probability matrix Π, the computation issue is significantly alleviated and we can easily obtain the closed-form expected error covariance for each time slot $\mathbb{E}[P_k|\mathbf{p},\mathbf{w}] = \sum_{i=1}^{T+1} \pi_{i,k} h^{i-1}(\overline{P})$.

The objective of the sensor is to minimize the trace of the expected average state estimation error covariance, i.e.,

$$J(\mathbf{p},\mathbf{w}) = \sum_{k=1}^{T} \mathrm{Tr}\left\{\mathbb{E}[P_k]\right\} = \sum_{k=1}^{T} \sum_{i=1}^{T+1} \pi_{i,k} h^{i-1}(\overline{P}) \qquad (16)$$

while the one of the attacker is to maximize it, under the energy constraints (10) and (11), respectively. Therefore, both sides constitute a two-player zero-sum game [7]. Denote $\mathbf{h} = [\overline{P}, h(\overline{P}), h^2(\overline{P}), \ldots, h^T(\overline{P})]'$, then (16) can be rewritten as $J(\mathbf{p},\mathbf{w}) = \mathbf{h}'\Pi\mathbf{1}$.

C. Equilibrium Analysis

From the perspective of the sensor, the sensor first assumes that the attacker adopts the strategy that maximizes the objective function and then makes its own decision, that is, the optimal action for the sensor is obtained by solving the following problem:

Problem 3.2: [The sensor's perspective]

$$\begin{aligned} &\min_{\mathbf{p}} \quad \max_{\mathbf{w}} \; J(\mathbf{p},\mathbf{w}) \\ &\text{s.t.} \quad \mathbf{p} \succeq 0, \quad \mathbf{w} \succeq 0, \quad \mathbf{1}'\mathbf{p} = \mathcal{P}, \quad \mathbf{1}'\mathbf{w} = \mathcal{W}. \end{aligned}$$

□

Solving Problem 3.2 is equivalent to solving two optimization problems under these constraints, which is well studied in the literature such as [9], [13], [20], and [21]. Denote the optimal value of Problem 3.2 as $J_s^\star$ and the associated optimal optimizers as $\mathbf{p}_s^\star$ and $\mathbf{w}_s^\star$.

Similarly, when considering the game from the perspective of the attacker, we just need to change the order of operations min and max in the objective function of Problem 3.2 with the same constraints.

Problem 3.3: [The attacker's perspective]

$$\begin{aligned} &\max_{\mathbf{w}} \quad \min_{\mathbf{p}} \; J(\mathbf{p},\mathbf{w}) \\ &\text{s.t.} \quad \mathbf{p} \succeq 0, \quad \mathbf{w} \succeq 0, \quad \mathbf{1}'\mathbf{p} = \mathcal{P}, \quad \mathbf{1}'\mathbf{w} = \mathcal{W}. \end{aligned}$$

□

We also have notations $J_a^\star$, $\mathbf{p}_a^\star$, and $\mathbf{w}_a^\star$ similar to the ones for the sensor. These two formulations from two perspectives are equivalent and, thus, the solutions make sense if and only if the two-player zero sum game admits a Nash equilibrium (NE) [22], [23]. Given the existence of an NE, we have $\max_{\mathbf{w}} \min_{\mathbf{p}} J(\mathbf{p},\mathbf{w}) = \min_{\mathbf{p}} \max_{\mathbf{w}} J(\mathbf{p},\mathbf{w})$, and $J_s^\star = J_a^\star, \mathbf{p}_s^\star = \mathbf{p}_a^\star, \mathbf{w}_s^\star = \mathbf{w}_a^\star$.

However, with continuous pure strategy sets, the set of mixed strategies is in the form of probability space over an infinite-dimensional set which is typically impossible to solve and implement in practice. This is the reason why we only investigate the existence of pure strategy Nash equilibrium, which depends on the following results [24].

Theorem 3.4: Consider a game with player index i, action sets $\mathbb{A}_i = \{a_i\}$, and objective functions $u(a_i, a_{-i})$, if for each player i:

1) $\mathbb{A}_i$ is compact and convex;
2) $u(a_i, a_{-i})$ is continuous in a_{-i};
3) $u(a_i, a_{-i})$ is continuous and concave in a_i;

then the game has a pure-strategy Nash equilibrium. □

Unfortunately, the objective function $J(\mathbf{p},\mathbf{w})$ is not concave, which means the game between the sensor and the attacker may not have a pure-strategy Nash equilibrium when they move simultaneously. Therefore, these two objective functions will lead to different solutions due to the well-known inequality [23] $\max_x \min_y f(x,y) \leqslant \min_y \max_x f(x,y)$ and we have $J_s^\star \geqslant J_a^\star, \mathbf{p}_s^\star \neq \mathbf{p}_a^\star, \mathbf{w}_s^\star \neq \mathbf{w}_a^\star$.

The difference between the optimal value and solutions is due to the advantage of assuming one player acts after predicting the opponent's strategy, that is, the sensor and the attacker move in a sequential order under a Stackelberg game framework, which is well studied in [20], [21], and [25]. Similar to the existing works, we summarize the aforementioned results in the following theorem.

Theorem 3.5: Consider the jamming game between the sensor and the attacker with continuous strategy sets:

1) the game has a mixed strategy Nash equilibrium in the form of probability space over an infinite-dimensional set;
2) the game does not always have a pure-strategy Nash equilibrium when both sides move simultaneously;
3) when the sensor moves after the attacker, the optimal strategy for the attacker $\mathbf{w}^\star$ is given by $\mathbf{w}_a^\star$, while the one for the sensor $\mathbf{p}^\star$ is given by

$$\begin{aligned} &\min_{\mathbf{p}} \quad J\left(\mathbf{p},\mathbf{w}_a^\star\right) \\ &\text{s.t.} \quad \mathbf{p} \succeq 0, \quad \mathbf{1}'\mathbf{p} = \mathcal{P} \end{aligned}$$

which is equal to $\mathbf{p}_a^\star$ (by definition) and the optimal value of the objective function for both sides is $J(\mathbf{p}_a^\star, \mathbf{w}_a^\star) = J_a^\star$;

4) when the attacker moves after the sensor, similarly we have the optimal strategy for the sensor $\mathbf{p}^\star$ given by $\mathbf{p}_s^\star$ while the one for the attacker $\mathbf{w}^\star$ is given by

$$\begin{aligned} &\min_{\mathbf{w}} \quad J\left(\mathbf{p}_s^\star,\mathbf{w}\right) \\ &\text{s.t.} \quad \mathbf{w} \succeq 0, \quad \mathbf{1}'\mathbf{w} = \mathcal{W} \end{aligned}$$

which is $\mathbf{w}_s^\star$ (by definition) and the optimal value of the objective function for both sides is $J(\mathbf{p}_s^\star, \mathbf{w}_s^\star) = J_s^\star$. □

Remark 3.6: Note that when the sensor moves first, we have the objective function as $J(\mathbf{p}_a^\star, \mathbf{w}_a^\star) = J_s^\star$. On the other hand, when the attacker moves first, we have the objective function $J(\mathbf{p}_s^\star, \mathbf{w}_s^\star) = J_a^\star$. As discussed before, we have $J_s^\star \geqslant J_a^\star$, that is, the sensor can benefit from moving later, and so does the attacker. □

The discussions in this section mainly focus on offline scenarios, that is, the problem formulation represents the situation when the capabilities of both sides are limited and have no access to the information of the real-time process, where they can only design their strategies before the process starts, independent of the state of the process. However, when the capabilities of both sides enable them to obtain additional information of the process, they can make decisions based on the current state and information collected from previous time steps. In such a situation, the framework in this section is not suitable anymore. Furthermore, in practice, most applications are designed for long-time running. These facts motivate us to further consider the scenario where both sides make interactive decisions through a two-player Markov decision process over an infinite time horizon.

IV. Infinite Time-Horizon: Markov Game Framework

When the capabilities of both sides enable them to obtain further information of the process (for example, when the sensor can receive the ACKs from the remote estimator indicating whether it has received the data packet successfully or not [17] at time k, the sensor can compute P_{k-1} using (9). The attacker can also eavesdrop the ACKs sent from the remote estimator to the sensor so that it can infer the state of the process), they can make decisions based on the current state and information collected from previous time steps. Since both sides also have dynamical information of their opponent by ACKs or other information, static analysis will be insufficient. In this section, we will set up the Markov game framework dealing with the dynamic jamming game to deal with such an interactive process over an infinite time horizon.

A. Preliminaries

To quantify the estimation quality over an infinite time-horizon, different from the last section and [13], we introduce a cost function J which quantifies the trace of the discounted sum of the expected state estimation error covariances as

$$J=\sum_{k=1}^{+\infty}\beta^k \mathrm{Tr}\left\{\mathbb{E}[P_k]\right\} \tag{17}$$

where $\beta\in[0,1)$ is the discount factor. The objective of the sensor is to help the remote estimator to obtain accurate state estimates $\hat{x}_k$. To be more specific, the sensor seeks to minimize J. On the other hand, the attacker tries to maximize the cost.

Both sides are subject to energy constraints, and the total energy constraints as in (10) and (11) are not suitable for the infinite-time horizon scenario. Instead, similar to [26], we take the energy consumption into consideration when we design the objective functions of the attacker and propose a more general one

$$J_A \triangleq \sum_{k=1}^{+\infty}\beta^k\left[\mathrm{Tr}\left\{\mathbb{E}[P_k]\right\}+\delta_s p_k-\delta_a\omega_k\right] \tag{18}$$

where $\delta_s,\delta_a\geqslant 0$ are weighting parameters, that is, the attacker aims to maximize the state estimation error covariance and minimizes its energy consumption. In addition, more energy resources consumed by the sensor are always preferred for the attacker. In contrast, the sensor has an opposite objective function

$$J_S \triangleq -J_A=\sum_{k=1}^{+\infty}\beta^k\left[-\mathrm{Tr}\left\{\mathbb{E}[P_k]\right\}-\delta_s p_k+\delta_a\omega_k\right]. \tag{19}$$

Thus, the sensor and the attacker involved in a two-player zero-sum game and the goals of both sides are the same: to maximize their respective objective functions.

In contrast to the offline design studied in the previous section, in the current situation, both sides take actions based on the information and process state at that time step, that is, they pursue an online design. To start, we first investigate the optimal strategy for the sensor when no attacker exists, and then extend our discussion to include both parties.

B. Optimal Strategy for the Sensor Without the Attacker

When no attacker exists ($\omega_k=0$), the objective function of the sensor in (19) can be modified with $\delta_a=0$ as

$$J_S=\sum_{k=1}^{+\infty}\beta^k\left[-\mathrm{Tr}\left\{\mathbb{E}[P_k]\right\}-\delta_s p_k\right]. \tag{20}$$

Denote P_{k-1} as the state of the process at time step k, which can take values in the state set $\mathbb{S}=\{\overline{P},h(\overline{P}),h^2(\overline{P}),\ldots\}$. Then, the reward for the sensor at time step k can be written as

$$r_k(P_{k-1},p_k)=\beta\left(-\mathrm{Tr}\left\{\mathbb{E}[P_k]\right\}-\delta_s p_k\right) \tag{21}$$

where, based on (7) and (9), $\mathbb{E}[P_k]=[1-2Q(\sqrt{\alpha(p_k/\sigma^2)})]\overline{P}+2Q(\sqrt{\alpha(p_k/\sigma^2)})h(P_{k-1})$.

Therefore, at each time step k, based on the process state P_{k-1}, the sensor chooses action $p_k=p_k(P_{k-1})$ and obtains the reward $r_k(P_{k-1},p_k)$ and moves to the next time step. Based on (7) and (9), suppose that the state at time step k is $P_{k-1}=h^i(\overline{P})$, $i\in\mathbb{N}$, then the state at time $k+1$, P_k, can only take two values $\overline{P}$ with probability $1-2Q(\sqrt{\alpha(p_k/\sigma^2)})$ and $h(P_{k-1})$ with probability $2Q(\sqrt{\alpha(p_k/\sigma^2)})$. Then the state transition probability is given by

$$t_k(P_k|P_{k-1},p_k)\triangleq\begin{cases}1-2Q\left(\sqrt{\alpha\frac{p_k}{\sigma^2}}\right), & \text{if } P_k=\overline{P}\\ 2Q\left(\sqrt{\alpha\frac{p_k}{\sigma^2}}\right), & \text{if } P_k=h(P_{k-1})\\ 0, & \text{otherwise.}\end{cases} \tag{22}$$

Note that since the reward function and transition probability function are stationary (time-invariant), we can replace $r_k(P_{k-1},p_k)$ and $t_k(P_k|P_{k-1},p_k)$ with $r(P_{k-1},p_k)$, $t(P_k|P_{k-1},p_k)$, respectively.

Define the transmission strategy of the sensor as $\pi_s=\{p_k(P_{k-1})\}$, then the objective function J_S is the expected sum of discounted one-stage rewards r_k under the strategy π_s with initial state $s\in\mathbb{S}$, that is, $J_S(s,\pi_s)=\sum_{k=1}^{+\infty}\beta^k\mathbb{E}[r_k(P_{k-1},p_k)]$, and the optimal value $J_S^\star(s)$ is $J_S^\star(s)=\arg\max_{\pi_s}J_S(s,\pi_s)$.

Given the framework discussed before, which can be regarded as a Markov decision process, the optimal value of the objective function $J_S^\star$ satisfies the following optimality (Bellman) equation [27]:

$$\begin{aligned}J_S^\star(s)=\max_{p_k}\big\{r(s,p_k)+\beta\,&\big[t(\overline{P}|s,p_k)J_S^\star(\overline{P})\\&+t\left(h(s)|s,p_k\right)J_S^\star\left(h(s)\right)\big]\big\}\end{aligned} \tag{23}$$

where s is the initial state taking values in $\mathbb{S}=\{\overline{P},h(\overline{P}),h^2(\overline{P}),\ldots\}$. Consequently, the optimal transmission strategy for the sensor is given by $p_k^\star(P_{k-1})=\arg\max_{p_k}\{r(P_{k-1},\,p_k)+\beta[t(\overline{P}|P_{k-1},p_k)J_S^\star(\overline{P})+t(h(P_{k-1})|P_{k-1},p_k)J_S^\star(h(P_{k-1}))]\}$.

Solving (23) is challenging in practice due to the complicated calculation of $J_{\mathbb{S}}^{\star}(s)$. There exist some well-known approaches to solve the problem, for example, value iteration and policy iteration [27]. These work well but require knowledge of the transition function and the reward for all states in the environment. However, the sensor and attacker may not have access to such information or carry out the computation. Therefore, to deal with the limitation of information or the computation issues, we obtain the optimal value through the approach of model-free reinforcement learning [28]–[30] as follows.

First, based on the right-hand side of (23), we can define $\mathcal{Q}(s, p_k) = r(s, p_k) + \beta[t(\overline{P}|s, p_k)J_{\mathbb{S}}^{\star}(\overline{P}) + t(h(s)|s, p_k)J_{\mathbb{S}}^{\star}(h(s))]$, where $\mathcal{Q}(s, p_k)$ represents the total discount reward with transmission power p_k and initial state s. Therefore, if we have the values of $\mathcal{Q}(s, p_k)$, then we can make an optimal decision simply based on

$$\begin{aligned} J_{\mathbb{S}}^{\star}(s) &= \max_{p_k} \mathcal{Q}(s, p_k) \\ p_k^{\star}(s) &= \arg\max_{p_k} \mathcal{Q}(s, p_k). \end{aligned} \tag{24}$$

Now the problem becomes calculating $\mathcal{Q}(s, p_k)$. We first initialize the values of $\mathcal{Q}(s, p_k)$ for all $s \in \mathbb{S}$ and transmission powers p_k with arbitrary values. Then, at each time step k with state s, we choose an action p_k based on (24), receive the corresponding reward $r_k(s, p_k)$ based on (21), move to another state s' randomly based on (9), and continue to the next time step. After each recursion, we update the corresponding Q-value based on the following equation (other Q-values remain the same):

$$\begin{aligned} \mathcal{Q}_{k+1}(s, p_k) &= (1 - \alpha_k)\mathcal{Q}_k(s, p_k) \\ &\quad + \alpha_k \left[r_k(s, p_k) + \beta \max_{p_k'} \mathcal{Q}_k\left(s', p_k'\right) \right] \end{aligned} \tag{25}$$

where $\alpha_k \in [0, 1)$ is the learning rate to be designed.

Remark 4.1: In this algorithm, the sensor can learn the optimal policy in an online pattern with a combination of learning and action. Clearly, a tradeoff between learning and action arises: on one hand, the learning can update the knowledge of the sensor about the process and the optimal policy to improve the performance; on the other hand, the random trials in the beginning will take time and affect the performance. Therefore, the idea of the Q-learning algorithm is to try different actions to update the knowledge when little information is available in the early time steps, and when the sensor has sufficient information about the process later, less emphasis is placed on the learning process. As a result, the learning rate needs to be designed to decay over the entire time horizon in order for the algorithm to converge to the optimal value $\mathcal{Q}(s, p_k)$ [28], [29]. This aspect will be illustrated in the simulation part. □

Remark 4.2: As in practice, the transmission power cannot grow infinitely, and we set limitations on the power which the sensor can choose: $p_k \in \mathbb{A}_s \triangleq [0, \overline{p}]$. As stated in [31], in power control architectures for cellular networks or other wireless networks, sending coarsely quantized power rather than actual power values is frequently used. To facilitate the algorithm, we discretize the power usage with L levels when we apply the Q-learning algorithm. When L is large, we can ignore the difference between the theoretical solution and the numerical one. □

Remark 4.3: Note that the state set is countable infinite. However, the occurrence probability of $h^i(\overline{p})$ is close to zero when i becomes sufficiently large. Therefore, for convenience, we can define a final state with $h^K(\overline{p})$ representing all of the states $h^i(\overline{p})$ with $i \geqslant K$, where K is a design parameter. □

The convergence rate of Q-learning for the discounted MDP is given by the following result.

Remark 4.4: Assume that the learning rate α_k takes the form of reciprocal of the times of the occurrence of the state-action pair visited, the following relations hold asymptotically and with probability one $|\mathcal{Q}_{k+1}(s, p_k) - \mathcal{Q}^{\star}(s, p_k)| \leqslant (C_0/k^{C_1(1-\beta)})$, and $|\mathcal{Q}_{k+1}(s, p_k) - \mathcal{Q}^{\star}(s, p_k)| \leqslant C_0\sqrt{\log\log k/k}$, where $C_0 > 0$ is a certain constant and C_1 is the maximal sampling probability of the state-action pair to minimal one ratio.

Proof: The main idea for proving this result is to compare $\mathcal{Q}_{k+1}(s, p_k)$ with a constructed simpler process which replaces $\mathcal{Q}_{k+1}$ with $\mathcal{Q}^{\star}$ in the updating process. The details are similar to [32] and we omit here. ■

By adopting this so-called Q-learning algorithm in (25), the sensor can learn the optimal transmission power strategy for the infinite time horizon in the case without the existence of the attacker based on the above discussion. The detailed convergence conditions will be provided in Theorem 4.6 in the following subsection.

C. Markov Sensor-Attacker Game Framework

After setting up the MDP framework for the sensor, we now consider the scenario with both the sensor and the attacker. At each time step k during the decision-making process, both the sensor and the attacker take actions and receive the reward (the corresponding state estimation error covariance and energy consumption at k), then move to the next stage modeled as a random process as described in (9).

To be more specific, the elements of this two-player Markov game are summarized as follows.

- *Player*: the sensor and the attacker.
- *State*: same as the case without the attacker, the state at time k is defined as $P_{k-1} \in \mathbb{S} = \{\overline{P}, h(\overline{P}), h^2(\overline{P}), \ldots\}$. We assume that the attacker can eavesdrop the ACKs sent from the remote estimator to the sensor so that it can infer the state of the process (otherwise, the attacker is similar to the offline game).
- *Action*: The action of the sensor is its transmission power $p_k \in \mathbb{A}_s = [0, \overline{p}]$. Similarly, the action of the attacker is the interference power $\omega_k \in \mathbb{A}_a \triangleq [0, \overline{\omega}]$.
- *Transition Probability*: similar to (22), define the state transition probability as $t_k(P_k|P_{k-1}, p_k, \omega_k)$

$$\begin{cases} 1 - 2Q\left(\sqrt{\alpha \frac{p_k}{\omega_k + \sigma^2}}\right), & \text{if } P_k = \overline{P} \\ 2\,Q\left(\sqrt{\alpha \frac{p_k}{\omega_k + \sigma^2}}\right), & \text{if } P_k = h(P_{k-1}) \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

- *Payoff*: The one-stage reward function for the sensor is defined as

$$r_k(P_{k-1}, p_k, \omega_k) \triangleq -\mathrm{Tr}\left\{\mathbb{E}[P_k]\right\} - \delta_s p_k + \delta_a \omega_k \tag{27}$$

and the reward function for the attacker is the opposite. The total discounted rewards for the for the sensor and for the attacker are J_S and J_A defined in (19) and (18), respectively.

Note that as the transition probability and the reward are stationary and independent of the time index, we can also write them as $t(P_k|P_{k-1}, p_k, \omega_k)$ and $r(P_{k-1}, p_k, \omega_k)$, respectively.

As we can see from before, the rewards and the state transition probability for the sensor not only depend on the sensor's own action p_k, but also on the action ω_k taken by the attacker, and vice-versa. To deal with such cases, the tools and frameworks developed in [13] with one-stage games, or single player's MDP discussed in Section IV-B are not sufficient. This motivates us to consider the Markov game framework [30], which is a generalization from a one-stage game or MDP to multiple-stage stochastic games.

Similar to the MDP and one-stage game, denote the strategy of the sensor and the attacker as $\pi_s = p_k(P_{k-1})$ and $\pi_a = \omega_k(P_{k-1})$, respectively. The objective function of the sensor with initial state $s \in \mathbb{S}$ under π_s and π_a are denoted as (the one for the attacker J_A is simply the opposite) $J_S(s, \pi_s, \pi_a) = \sum_{k=1}^{+\infty} \beta^k \mathbb{E}[r_k(P_{k-1}, p_k, \omega_k)]$.

Different from the optimal $J_S^\star$ under the MDP framework in the last part where the rewards only depend on the action taken by the sensor, in this two-player game, the player can adopt a Nash equilibrium strategy defined below, which has been proved in [29] as the best the player can do in the game.

Definition 4.5: In a two-player sensor-attacker stochastic game, a Nash equilibrium is a pair of strategies $(\pi_s^\star, \pi_a^\star)$ such that for all states $s \in \mathbb{S}$: $J_S^\star(s) \triangleq J_S(s, \pi_s^\star, \pi_a^\star) \geqslant J_S(s, \pi_s, \pi_a^\star)$, $\forall\, \pi_s$, and $J_A^\star(s) \triangleq J_A(s, \pi_s^\star, \pi_a^\star) \geqslant J_A(s, \pi_s^\star, \pi_a), \forall\, \pi_a$. □

To obtain the Nash equilibrium for this game, similar to the single-player Q-learning algorithm, we define the Q-value for the sensor in the Markov game as $\mathcal{Q}(s, p_k, \omega_k) = r(s, p_k, \omega_k) + \beta[t(\overline{P}|s, p_k, \omega_k) J_S^\star(\overline{P}) + t(h(s)|s, p_k, \omega_k) J_S^\star(h(s))]$, where $\mathcal{Q}(s, p_k, \omega_k)$ represents the total discount reward with transmission power p_k, interference power ω_k, and initial state s. Then, the optimal value $J_S^\star(s)$ can be solved by calculating the Nash equilibrium of the two-player Markov game with reward $Q(s, p_k, \omega_k)$.

When the sensor and the attacker have opposite objectives and rewards, which constitute a two-player zero-sum game as in our case, the calculation of the Nash equilibrium turns out to be a min-max problem as in [29], i.e.,

$$J_S^\star(s) = \max_{\pi_s} \min_{\pi_a} \sum_{p_k, \omega_k} \mathcal{Q}(s, p_k, \omega_k) \pi_s(p_k) \pi_a(\omega_k) \quad (28)$$

where the transmission strategy $\pi_s \in PD(\mathbb{A}_s)$. $PD(\mathbb{A}_s)$ represents the set of probability distributions over the set $\mathbb{A}_s$, and $\pi_s(p_k)$ is the probability of choosing the action p_k in strategy π_s. For the attacker, we have similar notations.

Now as before, the problem becomes calculating the value of $\mathcal{Q}(s, p_k, \omega_k)$. However, one cannot apply the Q-learning directly for each side independently without considering the action of its opponent as it may not work well when the opponent chooses a complex strategy, and the learning value may not converge. Therefore, we need a generalized version of the Q-learning for two-player game to overcome the limitation of the standard Q-learning. We first initialize the values of $\mathcal{Q}(s, p_k, \omega_k)$ for all $s \in \mathbb{S}$, transmission powers p_k and interference power ω_k with arbitrary values. Then, at each time step k with state s, we choose an action p_k and ω_k based on (30), receive the corresponding reward $r_k(s, p_k, \omega_k)$ based on (27), move to another state s' randomly based on (9), and continue to the next time step. After each recursion, we update the corresponding Q-value based on the following equation (other Q-values remain the same):

$$\begin{aligned}\mathcal{Q}_{k+1}(s, p_k, \omega_k) &= (1-\alpha_k)\mathcal{Q}_k(s, p_k, \omega_k) \\ &\quad + \alpha_k [r(s, p_k, \omega_k) + \beta \text{ Nash } \mathcal{Q}_k(s')] \end{aligned} \quad (29)$$

where

$$\text{Nash } \mathcal{Q}_k(s') \triangleq \max_{\pi_s} \min_{\pi_a} \sum_{p_k, \omega_k} \mathcal{Q}_k(s, p_k, \omega_k) \pi_s(p_k) \pi_a(\omega_k). \quad (30)$$

The convergence of the sequence (29) to the optimal values requires the following result.

Theorem 4.6: The Nash Q-learning sequence described in (29) will converge to the optimal value provided the following two conditions satisfied:

1) Every state $s \in S$ and action $a \in \mathbb{A}_s$ are visited infinitely often and the player only updates the Q-value corresponding to current state and actions.
2) The learning rate α_k satisfies $\alpha_k \in [0, 1)$, $\sum_{k=0}^{+\infty} \alpha_k = +\infty$, and $\sum_{k=0}^{+\infty} \alpha_k^2 < +\infty$. □

The proof of Theorem 4.6 is similar to the one in [29]. Note that these two conditions can be satisfied based on the discussions in Remark 4.1, 4.2, and 4.3. We will also illustrate how to implement them in the following simulation part.

Remark 4.7: As far as we know, the convergence rate of the general Nash Q-learning algorithm has not been investigated comprehensively in existing literature due to the uncertainty in the calculation of Nash equilibrium. However, in the case of zero-sum games, the convergence rate can be analyzed similarly to the Q-learning algorithm stated in Theorem 4.4. □

Remark 4.8: Note that typical methods for solving the Markov game require each side to have the knowledge of the information stated in the beginning of this subsection, that is, the system parameters, the state transition probability, and the closed-form expression for the total reward function J_A or J_S. However, though, we can calculate the state transition probability as in (26), the Q-learning method just requires the knowledge of system parameters and does not need such channel knowledge, such as the state transition probability, which provides a more practical way to solve the Markov game when the information is limited. Due to the wide adoption of real-time wireless communication between each part, the attacker and the operator will update their knowledge and adjust their strategies correspondingly. In these applications, the huge amount of channel information and the complex system model motivate us to consider the model-free reinforcement learning methods, such as Q-learning, to deal with these issues. □

Now we aim to investigate the stability of the process under such Markov games between the sensor and the attacker at equilibrium, in terms of the average state estimation error covariance over an infinite time horizon (or, equivalently, the asymptotic state estimation error covariance). Based on the recursion of P_k in (9) in the stationary state, we have $\limsup_{T\to\infty}(1/T)\sum_{k=1}^{T}\operatorname{Tr}(\mathbb{E}[P_k]) = \limsup_{k\to\infty}\operatorname{Tr}(\mathbb{E}[P_k]) = \sum_{i=0}^{+\infty}\pi_i \operatorname{Tr}(h^i(\overline{P}))$, $\sum_{i=0}^{+\infty}\pi_i = 1$, where $\pi_i \triangleq \mathbb{P}[P_k = h^i(\overline{P})] = (\prod_{j=0}^{i-1}(1-\epsilon_j))\epsilon_i$, $i > 0$, and $\epsilon_i \triangleq \mathbb{P}[\gamma_k = 1 | s = h^i(\overline{P})] = 1 - 2Q(\sqrt{\alpha(p_k^\star(s{=}h^i(\overline{P}))/\omega_k^\star(s{=}h^i(\overline{P}))+\sigma^2})$. Note that since there is a final state with $h^K(\overline{p})$ representing all of the states $h^i(\overline{p})$ with $i \geqslant K$ during the learning process, we only need to calculate ϵ_0 up to ϵ_K and $\epsilon_i = \epsilon_K$ for $i > K$.

Based on the properties of the Lyapunov operator $h(X) = AXA' + Q$ (e.g., [15]), a sufficient condition for the stability of the estimation process is given by

$$\min_{i=0,1,\ldots,K}\epsilon_i = 1-2Q\left(\sqrt{\alpha\frac{p_k^\star(s{=}h^i(\overline{P}))}{\omega_k^\star(s{=}h^i(\overline{P}))+\sigma^2}}\right) > \frac{1}{\rho(A)^2} \qquad (31)$$

where $\rho(A)$ is the spectral radius of A. Note that since the packet arrival rate is the same for $h^i(\overline{P})$ when $i \geq K$ (the transmission power strategies are the same for them), in (31), we only need to investigate up to K different arrival rates and require the minimal one to satisfy the condition.

Remark 4.9: Finding a weaker sufficient or necessary and sufficient conditions for the estimation process may be difficult since we do not know the optimal policies *a priori* and we cannot check this condition beforehand. This nontrivial problem is left as an important future direction of extension of our current work. □

Remark 4.10: As suggested by (31), when the attacker is quite powerful in terms of a dominant energy budget, which can be interpreted as a small energy weight σ_a (note that $\sigma_a = 0$ represents unlimited energy for the attacker), in addition to affect the estimation performance of the sensor, it may even jeopardize the stability of the system and cause severe consequences in many safety-critical CPS infrastructures. We provide a simulation example in the following section to illustrate this point. □

V. Numerical Example

In this section, we provide numerical examples to illustrate our results in different situations.

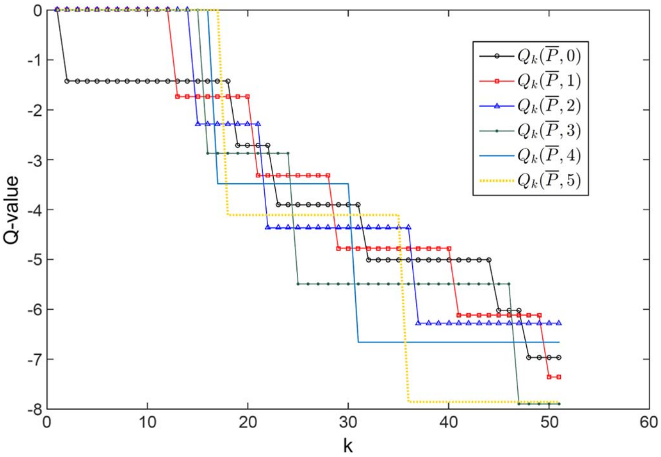


Fig. 2. Learning process for the optimal strategy of the sensor for state $\overline{P}$ (the first 50 time steps).

TABLE I
Converged Q-Values for the Sensor

	$Q_{100000}(s,p_k)$				
	$p_k = 0$	$p_k = 1$	$p_k = 2$	$p_k = 3$	$p_k = 5$
$s = \overline{P}$	**-50.8427**	-50.8486	-50.8640	-51.0501	-50.9902
$s = h(\overline{P})$	-52.2993	**-52.2986**	-52.3162	-52.3386	-52.4762
$s = h^2(\overline{P})$	-53.8753	**-53.7096**	-53.7097	-53.7284	-53.8204
$s = h^3(\overline{P})$	-55.9713	-55.3978	**-55.3917**	-55.3933	-55.4048
$s = h^4(\overline{P})$	-57.5775	-57.3249	-57.1922	-57.1648	**-57.1645**
$s = h^5(\overline{P})$	-61.1882	-59.1929	-59.4308	-59.0361	**-59.0300**
$s = h^6(\overline{P})$	-64.0566	-64.3578	-61.9343	-61.9049	**-61.6923**
$s = h^7(\overline{P})$	-66.9460	-62.4970	-62.4230	-61.8608	**-61.7006**

Consider the example with parameters of the system and the wireless network, where $A = 1.2$, $C = 0.7$, $Q = R = 0.8$, steady-state error covariance $\overline{P} = 0.9245$, network noise $\sigma^2 = 1$, and network parameter $\alpha = 3$. Now we apply the Q-learning algorithm in the infinite time-horizon Markov game. Suppose that $\delta_s = \delta_a = 1$, the discount factor $\beta = 0.96$, the learning rate $\alpha_k = 10/[15 + \text{ count }(s, p_k, \omega_k)]$ (based on Remark 4.1), where count (s, p_k, ω_k) is the times of the occurrence of the pair (s, p_k, ω_k). By designing the decay rate this way, it is easy to show that this rate satisfies the conditions in Theorem 4.6. As a consequence, the less visited state and action pairs will have more emphasis on the new learning knowledge (based on Remark 4.1). The recursion in (9) with a nonzero drop rate will also guarantee the visiting of every state and action in the simulations.

A. Q-Learning Without the Attacker

When the attacker does not exist, assume that $\overline{p} = 5$ with $L = 6$, that is, the power levels for the sensor are in the set $\{0, 1, 2, \ldots, 5\}$. Define $K = 8$, then the state set is $\{\overline{P}, h(\overline{P}), h^2(\overline{P}), \ldots, h^6(\overline{P}), h^7(\overline{P})(\text{and above})\}$. The learning process for the optimal strategy of the sensor is demonstrated in Fig. 2. (Only the first 50 time steps of the state $\overline{P}$ are shown for conciseness.) The initial value of all $\mathcal{Q}_0(s, p_k)$ are set to 0s and at each time step k, the corresponding Q-value is updated as (25) (other Q-values remain the same).

After 100 000 time steps iteration in Monte Carlo simulations, the Q-values converge to the optimal value as shown in Table I. The boldface value indicated the optimal Q-value $J_S^\star(s)$ for each state, therefore, the optimal strategy for the sensor is to use power level 0, 1, 1, 2, 5, 5, 5, 5 for the state $\overline{P}, h(\overline{P}), \ldots, h^6(\overline{P}), h^7(\overline{P})(\text{and above})$, respectively. The solution supports the intuition that when the error covariance at the remote estimator is relatively small, the sensor can use less energy for transmission. Otherwise, the sensor will choose a large energy level to increase the packet arrival rate and, therefore, improve estimation performance.

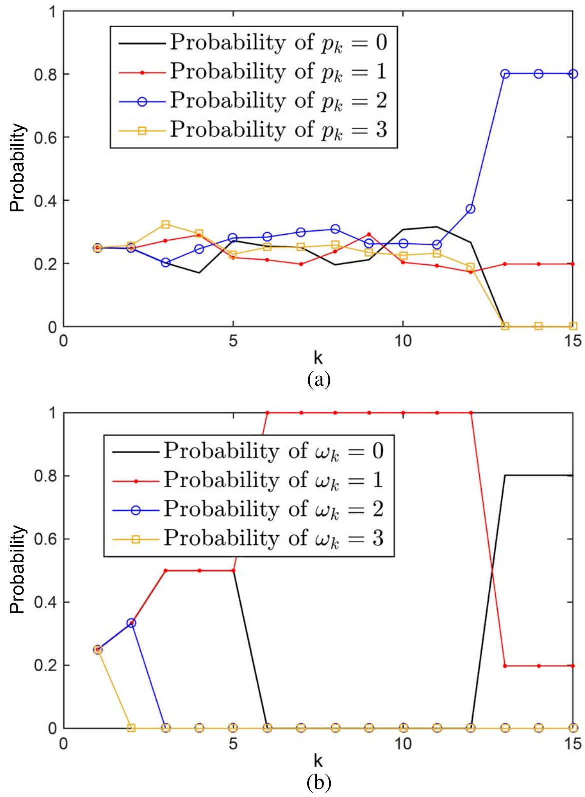


Fig. 3. Strategy learning process in state $\overline{P}$ for the sensor and the attacker (first 15 time steps). (a) Strategy learning process for the sensor in state $\overline{P}$ (first 15 time steps). (b) Strategy learning process for the attacker in state $\overline{P}$ (first 15 time steps).

Remark 5.1: As stated before, $\mathcal{Q}(s, p_k)$ represents the total discounted reward with transmission power p_k and initial state s. From Table I, we can see that when the initial state becomes large, which means the initial guess of the state is less accurate, the corresponding reward under the same action is reduced. This supports the intuition that the estimation performance is better with better initial estimation. □

B. Nash Q-Learning With the Attacker

When the attacker is involved in the game, both sides may adopt mixed strategies, that is, choose different actions based on different probabilities as in (28). Assume that $\overline{p} = \overline{\omega} = 3$ with $L = 4$ and $K = 4$, that is, the power levels for the sensor and the attacker are both {0, 1, 2, 3} and the state set is $\{\overline{P}, h(\overline{P}), h^2(\overline{P}), h^3(\overline{P}), h^4(\overline{P})(\text{and above})\}$. Since the Q-values of $\mathcal{Q}_k(s, p_k, \omega_k)$ have high dimension, to simplify the demonstration, we only show the learning process of the second state $\overline{P}$ as an example. As shown in Fig. 3(a) and (b), the sensor and the attacker adopt mixed strategy in the beginning, but the optimal strategy may be in the form of pure strategy: for example, the optimal strategy for the sensor when the state is $\overline{P}$, is to choose the power level $p_k = 1$ with probability 1 as shown in Fig. 4(b). As an illustration to Remark 4.10, Fig. 5 shows that when the attacker is quite powerful in terms of a dominant energy budget with $\sigma_a = 0.01$, the state estimation process diverges exponentially fast.

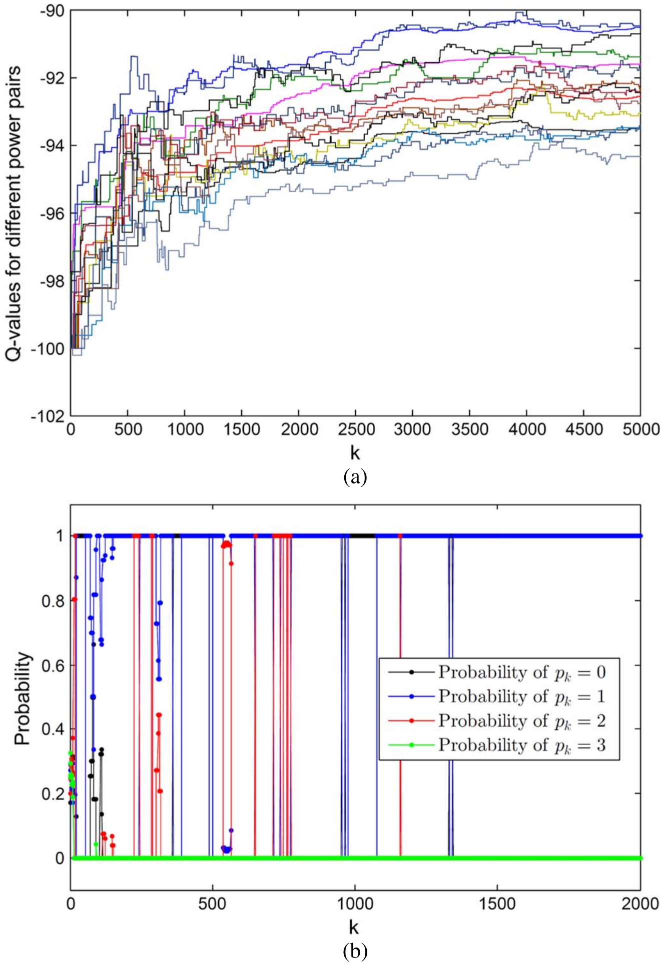


Fig. 4. Long-term learning process in state $\overline{P}$ for the sensor. (a) Q-values in state $\overline{P}$ for all 4 × 4 power-level combinations. (b) Convergence of strategy for the sensor in state $\overline{P}$.

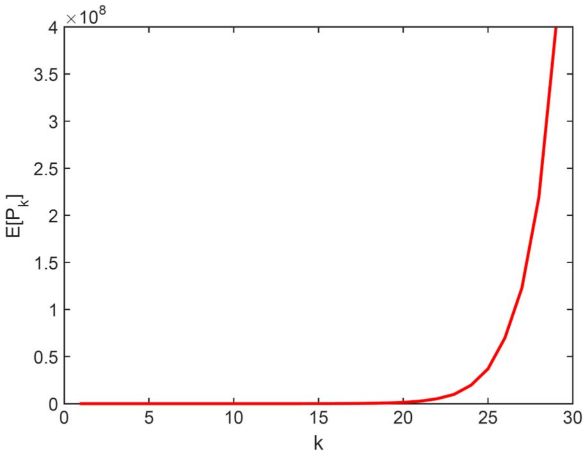


Fig. 5. Divergence of the state estimation process against a powerful (dominant energy budget) attacker.

VI. Conclusion

In this paper, we considered the associated games of remote state estimation in CPS using wireless links under SINR-based

DoS attacks. The two-player game when multiple power levels are available was first studied. Then, we built a Markov game framework to model the interactive decision-making process between the sensor and the attacker based on the current state of the process and information collected from previous time steps. To solve the corresponding optimality (Bellman) equations, we applied a modified Nash Q-learning algorithm. Numerical examples and simulations were provided to demonstrate our results.

References

[1] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.

[2] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[3] Y. Zhang, S. He, and J. Chen, "Data gathering optimization by dynamic sensing and routing in rechargeable sensor networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1632–1646, Jun. 2016.

[4] A. A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *Proc. IEEE 28th Int. Conf. Distrib. Comput. Syst. Workshops*, 2008, pp. 495–500.

[5] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal DoS attack scheduling in wireless networked control system," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 843–852, May 2016.

[6] T. Alpcan and T. Başar, *Network Security: A Decision and Game-Theoretic Approach*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[7] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. Philadelphia, PA, USA: SIAM, 1995, vol. 200.

[8] A. Gupta, C. Langbort, and T. Başar, "Optimal control in the presence of an intelligent jammer with limited actions," in *Proc. IEEE 49th Conf. Dec. Control*, 2010, pp. 1096–1101.

[9] H. Li, L. Lai, and R. C. Qiu, "A denial-of-service jamming game for remote state monitoring in smart grid," in *Proc. IEEE 45th Annu. Conf. Inf. Sci. Syst.*, 2011, pp. 1–6.

[10] S. Liu, P. X. Liu, and A. El Saddik, "A stochastic game approach to the security issue of networked control systems under jamming attacks," *J. Franklin Inst.*, vol. 351, no. 9, pp. 4570–4583, 2014.

[11] R. El-Bardan, S. Brahma, and P. K. Varshney, "Power control with jammer location uncertainty: A game theoretic perspective," in *Proc. IEEE 48th Annu. Conf. Inf. Sci. Syst.*, 2014, pp. 1–6.

[12] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal denial-of-service attack scheduling with energy constraint," *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 3023–3028, Nov. 2015.

[13] Y. Li, L. Shi, P. Cheng, J. Chen, and D. Quevedo, "Jamming attacks on remote state estimation in cyber-physical systems: A game-theoretic approach," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2831–2836, Oct. 2015.

[14] P. Hovareshti, V. Gupta, and J. S. Baras, "Sensor scheduling using smart sensors," in *Proc. IEEE 46th Conf. Dec. Control*, 2007, pp. 494–499.

[15] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.

[16] Y. Li, D. E. Quevedo, V. Lau, and L. Shi, "Optimal periodic transmission power schedules for remote estimation of ARMA processes," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6164–6174, Dec. 15, 2013.

[17] Y. Li, D. E. Quevedo, V. Lau, and L. Shi, "Online sensor transmission power schedule for remote state estimation," presented at the IEEE 52nd Annu. Conf. Dec. Control, Florence, Italy, 2013.

[18] J. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York, NY, USA: McGraw-Hill, 2007.

[19] L. Shi, M. Epstein, and R. M. Murray, "Kalman filtering over a packet-dropping network: A probabilistic perspective," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 594–604, Mar. 2010.

[20] A. Wang, Y. Cai, W. Yang, and Z. Hou, "A stackelberg security game with cooperative jamming over a multiuser ofdma network," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2013, pp. 4169–4174.

[21] R. Zhang, L. Song, Z. Han, and B. Jiao, "Physical layer security for two-way untrusted relaying with friendly jammers," *IEEE Trans. Veh. Technol.*, vol. 61, no. 8, pp. 3693–3704, Oct. 2012.

[22] J. Nash, "Non-cooperative games," *Ann. Math.*, vol. 54, no. 2, pp. 286–295, 1951.

[23] D. P. Palomar and Y. C. Eldar, *Convex Optimization in Signal Processing and Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[24] G. Debreu, "A social equilibrium existence theorem," *Proc. Nat. Acad. Sci. United States Amer.*, vol. 38, no. 10, pp. 886, 1952.

[25] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus, "Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2008, vol. 2, pp. 895–902.

[26] M. Adibi and V. T. Vakili, "Comparison of cooperative and non-cooperative game schemes for SINR-constrained power allocation in multiple antenna cdma communication systems," in *Proc. IEEE Int. Conf. Signal Process. Commun.*, 2007, pp. 1151–1154.

[27] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA, USA: Athena Scientific, 1995.

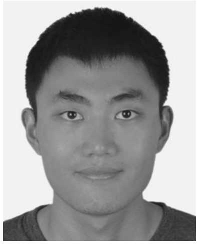
[28] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3/4, pp. 279–292, 1992.

[29] J. Hu and M. P. Wellman, "Multiagent reinforcement learning: Theoretical framework and an algorithm," in *Proc. ICML*, 1998, vol. 98, pp. 242–250.

[30] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. ICML*, 1994, vol. 94, pp. 157–163.

[31] D. E. Quevedo, A. Ahlén, and J. Østergaard, "Energy efficient state estimation with wireless sensors through the use of predictive power control and coding," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4811–4823, Sep. 2010.

[32] C. Szepesvári *et al.*, "The asymptotic convergence-rate of q-learning," in *Proc. NIPS*, 1997, pp. 1064–1070.



Yuzhe Li (M'09) received the B.S. degree in mechanics from Peking University, Peking, China, in 2011 and the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Hong Kong, China, in 2015.

In 2013, he was a Visiting Scholar in the University of Newcastle, Australia. He is currently a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, University of Alberta, Canada. His current research interests include cyber-physical system security, sensor power control, and networked state estimation.

Daniel E. Quevedo (SM'14) received the Ing. Civ. Electrónico and M.Sc. degrees from Universidad Técnica Federico Santa María, Chile, in 2000 and 2005, respectively, and the Ph.D. degree from the University of Newcastle, Australia.

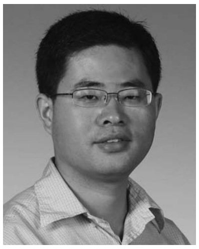
He is Editor of the *International Journal of Robust and Nonlinear Control* and serves as Chair of the IEEE CSS *Technical Committee on Networks & Communication Systems* and is Chair of Automatic Control at Paderborn University, Paderborn, Germany. His research mainly focuses on networked estimation and control and on predictive control of power converters.

Prof. Quevedo was awarded a five-year fellowship from the Australian Research Council in 2009.

Subhrakanti Dey (SM'06) received the B.Tech. and M.Tech. degrees from the Indian Institute of Technology, Kharagpur, India, in 1991 and 1993, respectively, and the Ph.D. degree from Australian National University, Canberra, Australia, in 1996.

Currently, he is a Professor with the Department of Engineering Sciences at Uppsala University, Uppsala, Sweden. His current research interests include networked control systems, wireless communications and networks, signal processing for sensor networks, as well as stochastic and adaptive signal processing and control.

Prof. Dey currently serves on the Editorial Board of IEEE TRANSACTIONS ON SIGNAL PROCESSING and IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS. He was also an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING during 2007–2010 and the IEEE TRANSACTIONS ON AUTOMATIC CONTROL during 2004–2007, and Associate Editor for *Systems and Control Letters* during 2003–2013.



Ling Shi (M'08) received the B.S. degree in electrical and electronic engineering from Hong Kong University of Science and Technology, Hong Kong, China, in 2002 and the Ph.D. degree in control and dynamical systems from the California Institute of Technology, Pasadena, CA, USA, in 2008.

Currently, he is an Associate Professor with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology. He has been a subject editor for the *International Journal of Robust and Nonlinear Control* since 2015. He was also an Associate Editor for a special issue on Secure Control of Cyber Physical Systems in the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS in 2015. His research interests include networked control systems, wireless-sensor networks, event-based state estimation and sensor scheduling, and smart energy systems.