

Performance and Resilience of Cyber-Physical Control Systems with Reactive Attack Mitigation

Subhash Lakshminarayana *Member, IEEE*, Jabir Shabbir Karachiwala, Teo Zhan Teng, Rui Tan *Senior Member, IEEE*, and David K.Y. Yau *Senior Member, IEEE*

Abstract—This paper studies the performance and resilience of a linear cyber-physical control system (CPCS) with attack detection and reactive attack mitigation in the context of power grids. It addresses the problem of deriving an optimal sequence of false data injection attacks that maximizes the state estimation error of the power system. The results provide basic understanding about the limit of the attack impact. The design of the optimal attack is based on a Markov decision process (MDP) formulation, which is solved efficiently using the value iteration method. We apply the proposed framework to the voltage control system of power grids and run extensive simulations using PowerWorld. The results show that our framework can accurately characterize the maximum state estimation errors caused by an attacker who carefully designs the attack sequence to strike a balance between the attack magnitude and stealthiness, due to the simultaneous presence of attack detection and mitigation. Moreover, based on the proposed framework, we analyze the impact of false positives and negatives in detecting attacks on the system performance. The results are important for the system defenders in the joint design of attack detection and mitigation to reduce the impact of these attack detection errors. Finally, as MDP solutions are not scalable for high-dimensional systems, we apply Q-learning with linear and non-linear (neural networks based) function approximators to solve the attacker’s problem in these systems and compare their performances.

Keywords: Cyber-physical control system, power grid voltage control, Markov decision process, Q-learning, Neural networks

I. INTRODUCTION

Critical infrastructures such as power grids are increasingly becoming the target of cyber attacks as evidenced in recent high-profile incidents, e.g., the BlackEnergy attack [2]. These attacks injected false sensor data and/or control commands to the industrial control systems and resulted in widespread damage to the physical infrastructures and service outages. These incidents alert us to a general class of attacks called *false data injection* (FDI) against cyber-physical systems (CPS).

S. Lakshminarayana is with the University of Warwick, UK (email: subhash.lakshminarayana@warwick.ac.uk). J.S. Karachiwala is with Oracle India Pvt. Ltd. (email: jabirrokx@yahoo.com). Z. Teng is with GovTech Singapore (email: teozt@hotmail.com). R. Tan is with the Nanyang Technological University, Singapore (email: tanrui@ntu.edu.sg). D.K.Y. Yau is with the Singapore University of Technology and Design, Singapore (email: david_yau@sutd.edu.sg).

This research was supported in part by the National Research Foundation (NRF), Prime Minister’s Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-31), in part by the the Energy Programme and administered by the Energy Market Authority (EP award no. NRF2017EWT-EP003-06), in part by SUTD-ZJU IDEA award no. (VP) 201805, and in part by a start-up grant at NTU. We acknowledge the support of NVIDIA Corporation with the donation of two GPUs used in this research. We also acknowledge Zhenyu Yan for managing the computation resources used in this paper.

The paper was presented in part at the ACM Int’l Conf. Future Energy Systems (e-Energy), 2017 [1].

Attack detection and mitigation are two basic CPS security research problems, where the *attack detection* makes decisions in real time regarding the presence of an attack and *attack mitigation* isolates a detected attack and/or reduces its adverse impact on the system performance. CPSs often have various built-in anomaly detection methods that are effective in detecting simple fault-like FDI attacks, such as injecting surges, ramps, and random noises. However, critical CPSs (e.g., power grids) are the target of sophisticated attackers (such as hostile national organizations), whose attacks are often well-crafted using detailed knowledge of the system and its anomaly detection methods. To avoid detection, the attacker can inject a sequence of attacks of small magnitude and gradually mislead the system to a sub-optimal and even unsafe state. However, due to the stochastic nature of the physical and measurement processes of CPSs, as well as the adoption of stringent, advanced attack detectors, the well-crafted attacks can be detected probabilistically [3]. Upon the detection, mitigation should be activated to isolate the attack or maintain acceptable system performance in coexisting with the attack.

Therefore, attack detection and mitigation are deeply coupled and they jointly define the system resilience against FDI attacks. On the one hand, a conservative detector may miss attacks, causing system performance degradation due to the mis-activation of attack mitigation. On the other hand, an aggressive detector may frequently raise false positives, triggering unnecessary mitigation actions in the absence of attacks, while attack mitigation generally needs to sacrifice the system performance to increase its robustness against attacks. Thus, it is important to understand the joint effect of attack detection and mitigation on the system performance, which serves as a basis for designing satisfactory detection-mitigation mechanisms. However, prior research on FDI attacks mostly study attack detection and mitigation separately [4], [3], [5], and falls short of capturing their joint effect on the system. The studies on attack detection [4], [3] generally ignore the attack mitigation triggered by probabilistic detection of attacks, and its impact on the future system states. On the other hand, the studies on attack mitigation [6], [7] assume that the attack has been detected, and ignore the probabilistic nature of the attack detection and any adverse impact of mis-activation or false activation of mitigation due to missed detections (MDs) and false positives (FPs).

As an early (but important) effort in closing the gap, we jointly consider attack detection and mitigation in the system defense and study their joint effect from both the attacker and the defender’s perspectives. In particular, from the attacker’s perspective, we investigate the largest system performance

degradation that a sophisticated attacker can cause in the presence of such a detection-mitigation defense mechanism. Studying this largest performance degradation helps us quantify the limit of attack impact. From the defender’s perspective, we quantify the system performance degradation due to FPs and MDs. The framework serves as an important basis for designing and assessing attack detection and mitigation strategies for critical infrastructures.

Although our analysis applies to general control systems, the primary focus of this work is power grid controls. We consider a general discrete-time linear time invariant (LTI) system with a feedback controller that computes its control decision based on the system state estimated by a Kalman filter (KF). The LTI model is a widely used approximation to model many control loops in a power grid, such as the power grid’s voltage control system [8], [9] and the generator swing equations [10]. Moreover, under the assumption of direct current (DC) power flows, the observation mode can also be linearized [11], [12]. These LTI models have been widely used in power grid research literature [13], [14], [15]. For each time step, the controller uses a χ^2 attack detector [16], and activates mitigation actions upon detecting an attack. Following the Kerckhoffs’s principle, we consider an attacker who accurately knows the system and its attack detection and mitigation methods. The attacker launches FDI attacks on the sensor measurements over an attack time horizon, aiming at misleading the controller into making erroneous control decisions.

As the attack detection at each time step is probabilistic, we formulate the attacker’s problem as a constrained stochastic optimization problem with an objective of maximizing the state estimation error over the attack time horizon, subject to a general constraint that the energy of the attack signal is upper-bounded. The attacker faces a fundamental dilemma in designing his attack – a large attack magnitude will result in high detection probability, thus nullifying the attack impact on the system (due to mitigation) whereas a small attack magnitude increases stealthiness but may cause little damage. Thus, the solution to this problem leads to an attack sequence that strikes a balance between attack magnitude and stealthiness to achieve the largest system performance degradation.

The main challenge in solving the aforementioned attacker’s problem lies in the fact that the system state at any time depends on all the past attack detection results, due to reactive attack mitigation. Thus, the optimal attack at any time must exhaustively account for all possible sequences of past detection results, which is computationally complex. Moreover, the probabilistic attack detection introduces additional randomness into the system dynamics. Our key observation to overcome these issues is that the system dynamics is Markovian and the attacker’s injections at any time can be computed based on the knowledge about it, which captures the impact of all the past detection results. To summarize, the main contributions of our work are as follows:

- We solve the aforementioned attacker’s problem using a Markov decision process (MDP) framework. In our formulation, the sequential operations of probabilistic attack detection and mitigation are mapped to the MDP’s state transition

probabilities. The MDP is solved by state space discretization and using the *value iteration* algorithm [17].

- However, the value iteration algorithm is computationally expensive, especially for high-dimensional systems. To address this issue, we apply a Q-learning with linear and non-linear (neural networks based) function approximation method to solve the attacker’s problem in these systems.

- Based on the above framework, we consider the problem of designing the detection threshold from the defender’s perspective. To this end, we derive analytical expressions to quantify the cost of FPs and MDs. Based on these cost functions, the attack detection threshold can be tuned to balance the effects of FPs and MDs depending on the accuracy of the mitigation signal.

- We apply our framework to the power grid voltage control that regulates the pilot bus voltages by adjusting the generators’ reactive power outputs. The attacker injects false voltage measurements, aiming at deviating the pilot bus voltages. Extensive simulations using PowerWorld show that the optimal attack sequence computed using our approach causes the maximum deviations of the pilot bus voltages.

The rest of the paper is organized as follows. Section II reviews related work. Section III describes the system model. Section IV formulates the research problem. Section V describes the MDP-based solution. Section VI analyzes the impact of FPs and MDs on the system performance. Section VIII presents the simulation results. Section IX concludes.

II. RELATED WORK

Most of the existing studies treat attack detection and mitigation problems separately. In the category of attack detection, the performance degradation caused by stealthy attacks in a noiseless LTI system has been analyzed [18]. However, non-determinism and measurement noises experienced by real-world systems provide an opportunity for the attacker to masquerade his attack as natural noises, thereby rendering attack detection probabilistic. For a general LTI system, research [3] and [4] studied the impact of stealthy FDI attacks, and derived optimal attack sequences that can cause the worst system performance degradation. The impact of such stealthy FDI attacks on system efficiency and safety in the context of power grids have also been investigated [19], [20], [21]. In particular, the economic impact of FDI attacks were studied in [19] and [20]. Reference [21] showed that the attacker can drive the power system frequency to unsafe levels by injecting a sequence of carefully-crafted FDI attacks. However, all these studies [4], [3], [5], [19], [20], [21] ignore the attack mitigation triggered by probabilistic detection of attacks and its impact on the future system states and attack detection.

In the category of attack mitigation, preventive and reactive mitigation strategies have been proposed. Preventive mitigation identifies vulnerabilities in the system design and removes them to prevent exploitation by attackers. For instance, in a power grid, a set of sensors and their data links can be strategically selected and protected such that a bad data detection mechanism cannot be bypassed by FDI attacks against other unprotected sensors and links [22], [23], [24].

However, preventive mitigation provides static solutions only, which do not address the adaptability of strategic attackers against critical infrastructures. Thus, in addition to preventative mitigation, it is important to develop reactive attack mitigation, i.e., countermeasures that are initiated after detecting an attack. Reactive attack mitigation is mainly studied under game-theoretic settings, both in the context of general LTI systems [6], [7] and power grids [25], [26], [27]. In particular, [25] and [26] studied the mitigation of FDI attacks against power grids as one-shot games, where as [27] formulated the repeated interactions between the attacker and the defender's actions under a Markov game framework. However, the aforementioned studies on reactive mitigation assume that the attack has been detected, and ignore the impact of uncertain attack detection on the overall attack mitigation. In contrast, our framework captures the interdependence between the attack detection and mitigation, and their joint impact on the system's dynamics and performance.

III. PRELIMINARIES

The analysis in this paper is based on the general discrete-time LTI model. Using this LTI model, we can develop a general framework that applies to many control loops in the power grid (please refer to Section III-A1), rather than being specific to individual control loops and analyzing each of them separately.

A. System Model

A block diagram of the system model is illustrated in Fig. 2. We consider a general discrete-time LTI system evolving as

$$\mathbf{x}[t+1] = \mathbf{A}\mathbf{x}[t] + \mathbf{B}\mathbf{u}[t] + \mathbf{w}[t], \quad (1)$$

where $\mathbf{x}[t] \in \mathbb{R}^n$ is the system state vector, $\mathbf{u}[t] \in \mathbb{R}^p$ is the control input, and $\mathbf{w}[t] \in \mathbb{R}^n$ is the process noise at the t -th time slot. Matrices \mathbf{A} and \mathbf{B} denote the propagation and control matrices, respectively. The initial system state $\mathbf{x}[0]$ and process noise $\mathbf{w}[t]$ are independent Gaussian random variables. Specifically, $\mathbf{x}[0] \sim \mathcal{N}(\mathbf{0}, \mathbf{X})$ and $\mathbf{w}[t] \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$, where $\mathbf{0} = [0, \dots, 0]^T$ and \mathbf{X} and \mathbf{Q} are the covariance matrices. The process described in (1) is observed through sensors deployed in the system, whose observation at time t , denoted by $\mathbf{y}[t] \in \mathbb{R}^m$, is given by

$$\mathbf{y}[t] = \mathbf{C}\mathbf{x}[t] + \mathbf{v}[t], \quad (2)$$

where $\mathbf{C} \in \mathbb{R}^{m \times n}$ is the measurement matrix and $\mathbf{v}[t] \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is the measurement noise at time t with a covariance of \mathbf{R} . We assume that $\mathbf{v}[t]$ is independent of $\mathbf{x}[0]$ and $\mathbf{w}[t]$. Moreover, we assume that the system in (1) is controllable and the measurement process in (2) is observable.

From a practical point of view, the process noise models the mismatch between the actual system state (following a control action) and that predicted by the linear model in (1). The observation noise is due to the sensor measurement noise.

In the context of power grids, these noises can be modeled as Gaussian random variables¹.

1) *LTI Systems in Power Grids:* We provide examples of a discrete-time LTI system, namely a power system's voltage control and generator swing equations. A power system consists of a set of buses (nodes) to which generators and loads are connected, and transmission lines connecting these buses. Fig. 1 illustrates the IEEE 9-bus test system.

Voltage control: Power system voltage control refers to maintaining the voltages of selected critical buses (called *pilot buses* marked with "P" in Fig. 1) within safe operational limits by adjusting the output voltage of the generator buses [8], [9]. It can be modeled as an LTI system described in Eqs. (1) and (2). Specifically, the state vector $\mathbf{x}[t]$ refers to the voltages of the pilot buses at time t , which should be maintained at a nominal voltage denoted by \mathbf{x}_0 . The control signal, which is applied at the generator buses, corresponds to the change in the generator bus voltages, i.e., $\mathbf{u}[t] = \mathbf{v}_G[t] - \mathbf{v}_G[t-1]$, where $\mathbf{v}_G[t]$ is a vector of the generator bus voltages. We clarify here that we do not consider the detailed model of the generator control loop at the generator bus explicitly. Instead, the control actions in our model only considers the changes in the voltage of the generator bus, while abstracting away finer details of the generator control loop at the generator buses.

Under this model, the voltage control system can be approximated by an LTI system with $\mathbf{A} = \mathbf{I}$ [9],[8]. The control matrix \mathbf{B} models the dependency between change of voltage at the generator bus (i.e., the control signal) and the change of voltage at the pilot buses (i.e., the state). This relationship depends on the power flow within the network. Note that in our work, instead of using the power flow equations to model \mathbf{B} (e.g., by ac or dc power flow equations), we used a data-driven approach in which we estimate the matrix \mathbf{B} from data traces obtained in a PowerWorld simulation. More details on estimating the matrix \mathbf{B} will be presented in Section VIII. Since the estimation cannot be perfect, the LTI model may be inaccurate, though the inaccuracies are small and can be captured as process noise. As the system state can be directly measured by voltage sensors (e.g., phasor measurement units, PMUs) deployed at the pilot buses, the measurement matrix is an identity matrix, i.e., $\mathbf{C} = \mathbf{I}$. The system is bounded-input bounded-output stable if the control algorithm satisfies $\mathbf{B}\mathbf{u}[t] = \alpha(\mathbf{x}_0 - \mathbf{x}[t])$ for $\alpha \in (0, 1)$, and this control is adopted in practical systems [8]. However, as the sensor measurements are noisy, the controller cannot have perfect knowledge of the system state $\mathbf{x}[t]$. Rather, the state is estimated using the KF-based technique described in (3). Based on the estimated state $\hat{\mathbf{x}}[t]$, the control can be computed as $\mathbf{u}[t] = \alpha\mathbf{B}^{-1}(\mathbf{x}_0 - \hat{\mathbf{x}}[t])$.

Generator swing equations: The swing equations establish a mathematical relationship between the angles of the mechanical motor and the generated alternating current electricity [10]. The swing equations can be linearized and modeled as an LTI system described by Eqs. (1) and (2) under the assumption of

¹Although our analysis assumes Gaussian noise, in Section VIII, we perform simulations with non-Gaussian noises and show the performance of the algorithms developed under this setting. The non-Gaussian setting may be important to model real-world sensor measurement noises, for instance, noises from phasor measurement units (PMUs) [28].

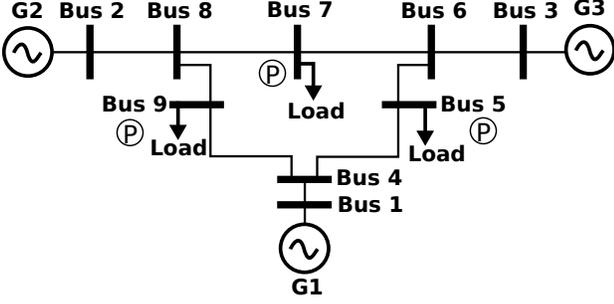


Fig. 1. IEEE 9-bus power system.

direct current (DC) power flow. For a power network consisting of n generators, the state vector consists of $2n$ entries. The first n entries are the generator's rotor phase angles and the last n entries are the generator's rotor frequency. The control inputs correspond to changes in mechanical input power to the generators, and is responsible for maintaining the generator's rotor angle and frequency within a safe operational range. The entries of the matrix \mathbf{A} depend on the power system's topology (including the transmission lines' susceptances) as well as the generators' mechanical parameters (such as inertia and damping constants). The structure of the matrix \mathbf{B} depends on the type of feedback control used to restrict the rotor angle frequency to within the safety range [10]. The measurement vector $\mathbf{y}[t]$ under the DC power flow model includes nodal real power injections at all the buses, all the branch power flows, and the rotor angles. The observation matrix \mathbf{C} can be constructed based on the power system topology [11], [12]. These linearized control models have been widely used in power grid research literature [13], [14], [15].

2) *Power Grids Dynamic State Estimation Using Kalman Filter*: We consider Kalman filter (KF), which is widely used for state estimation of a dynamic system, and adopted in power grids' transmission [29] and distribution grids [30] as well. The KF works as follows:

$$\hat{\mathbf{x}}[t+1] = \mathbf{A}\hat{\mathbf{x}}[t] + \mathbf{B}\mathbf{u}[t] + \mathbf{K}(\mathbf{y}[t+1] - \mathbf{C}(\mathbf{A}\hat{\mathbf{x}}[t] + \mathbf{B}\mathbf{u}[t])), \quad (3)$$

where $\hat{\mathbf{x}}[t]$ and $\hat{\mathbf{x}}[t+1]$ are the estimates of the system states at the t -th and $t+1$ -th time slots, respectively, \mathbf{K} denotes the steady-state Kalman gain given by $\mathbf{K} = \mathbf{P}_\infty \mathbf{C}^T (\mathbf{C} \mathbf{P}_\infty \mathbf{C}^T + \mathbf{R})^{-1}$, and the matrix \mathbf{P}_∞ is the solution to the algebraic Riccati equation $\mathbf{P}_\infty = \mathbf{A} \mathbf{P}_\infty \mathbf{A}^T + \mathbf{Q} - \mathbf{A} \mathbf{P}_\infty \mathbf{C}^T (\mathbf{C} \mathbf{P}_\infty \mathbf{C}^T + \mathbf{R})^{-1} \mathbf{C} \mathbf{P}_\infty \mathbf{A}^T$. We denote the KF estimation error at time t by $\mathbf{e}[t] = \mathbf{x}[t] - \hat{\mathbf{x}}[t]$.

Note that in this work, we focus on the estimation of a sequence of steady states of the power system. Compared to single snapshot-based weighted least square (WLS) method (i.e., the traditional state estimation algorithm [11]), dynamic state estimators like Kalman filter exploit the valuable historical information of the system states to reduce the computational complexity involved in computing the state estimate (i.e., observe that the evolution of the state estimate is linear (3)).

B. Attack Model, Detection, and Mitigation

1) *Attack Model*: Modern-day critical infrastructure systems extensively use ICT for their operation. For instance, in a power grid, the remote terminal units (RTUs) and many other field devices are connected by the Internet protocol (IP) [31]. Additionally, in a power grid, the sensors (such as the voltage and current measurement units) are spread over a large geographical area, making their measurements vulnerable to physical attacks. Such vulnerabilities can be exploited to launch attacks and disrupt the normal power grid operations.

In this paper, we follow Kerckhoffs's principle and consider an attacker who has accurate knowledge of the targeted CPCS and read access to the system state. Such knowledge can be obtained in practice by malicious insiders, long-term data exfiltration [32], or social engineering against employees, contractors, or vendors of a critical infrastructure operator [33]. Specifically, we assume that the attacker knows the matrices \mathbf{A} , \mathbf{B} and \mathbf{C} , as well as the operational details of the KF and the system's method of anomaly detection (including the detection threshold). In addition, the attacker also has read and write accesses to the system measurements, e.g., using PLC rootkits [34].

We consider FDI attacks on the system sensors. The compromised observations, denoted by $\mathbf{y}_a[t]$, are given by $\mathbf{y}_a[t] = \mathbf{y}[t] + \mathbf{a}[t]$, where $\mathbf{a}[t] \in \mathbb{R}^m$ is the attacker's injection. To model the attacker's energy constraint, we assume that the norm of the injection, $\|\mathbf{a}[t]\|$, is upper-bounded by a constant a_{\max} , i.e., $\|\mathbf{a}[t]\| \leq a_{\max}$. Denote by \mathcal{A} the set of all feasible attack vectors that satisfy the above energy constraint.

2) *Attack Detection and Mitigation*: We assume that the controller uses the χ^2 detector [16] to detect the attack, which has been widely adopted in security analysis of LTI systems [4], [3]. The χ^2 detector computes a quantity $g[t] = \mathbf{r}[t]^T \mathbf{P}_r^{-1} \mathbf{r}[t]$, where $\mathbf{r}[t]$ is the residual given by $\mathbf{r}[t+1] = \mathbf{y}_a[t+1] - \mathbf{C}(\mathbf{A}\hat{\mathbf{x}}[t] + \mathbf{B}\mathbf{u}[t])$ and $\mathbf{P}_r = \mathbf{C} \mathbf{P}_\infty \mathbf{C}^T + \mathbf{R}$ is a constant matrix that denotes the covariance of the residual in the steady state. The detection result of the detector is denoted by $i[t] \in \{0, 1\}$. The detector declares an attack if $g[t]$ is greater than a predefined threshold η . Specifically, if $0 \leq g[t] \leq \eta$, $i[t] = 0$; otherwise, $i[t] = 1$.

Based on the detection result, the controller applies a reactive mitigation action. If the χ^2 detector's alarm is triggered, the controller forwards a modified version of the observation $\mathbf{y}_a[t] - \delta[t]$ to the KF, where $\delta[t] \in \mathbb{R}^m$ is an attack mitigation signal; otherwise, the controller directly forwards $\mathbf{y}_a[t]$ to the KF (ref. Fig. 2). Thus, the controller's operation is

$$\mathbf{y}_f[t] = \mathbf{y}_a[t] - i[t]\delta[t]. \quad (4)$$

With the controller's mitigation action, the KF estimate is

$$\hat{\mathbf{x}}[t+1] = \mathbf{A}\hat{\mathbf{x}}[t] + \mathbf{B}\mathbf{u}[t] + \mathbf{K}(\mathbf{y}_f[t+1] - \mathbf{C}(\mathbf{A}\hat{\mathbf{x}}[t] + \mathbf{B}\mathbf{u}[t])). \quad (5)$$

The mitigation signal $\delta[t]$ can be generated using existing mitigation approaches [35], [36], [37], [38]. Specifically, once the attack is detected, a conservative mitigation approach is to ignore the sensor measurements completely, and drive the system based only on the model. For instance, the estimated state

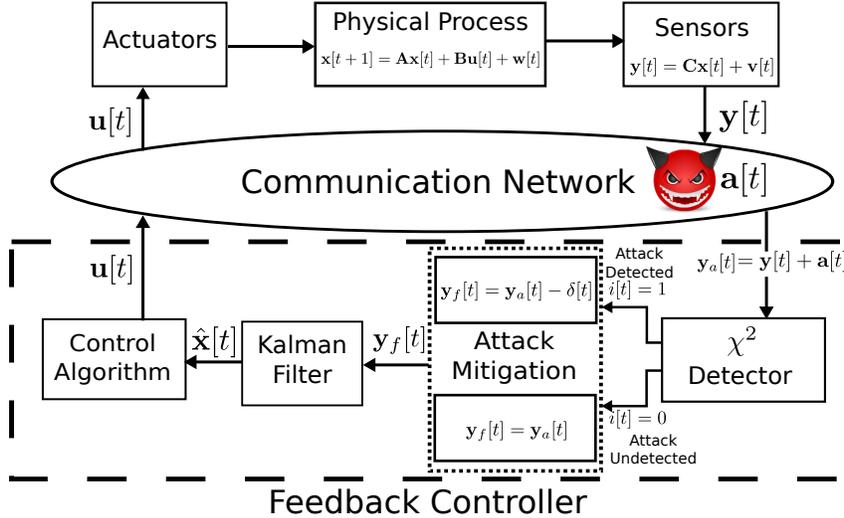


Fig. 2. Block diagram of the system model.

$\hat{\mathbf{x}}[t+1] = \mathbf{A}\hat{\mathbf{x}}[t] + \mathbf{B}\mathbf{u}[t]$ can be which is subsequently used to compute the control decisions. Alternately, upon detecting an attack, the controller can ignore the sensor measurements, and instead make an *educated guess* to obtain the true sensor measurements [36]. For instance, in power grids, accurate real-time load forecasts can be computed using techniques such as regression models, neural networks and statistical learning algorithms. The forecasted measurements can be used to drive the Kalman filter estimates. The operator can also run a secure state estimation algorithm (e.g., [37], [38]) to obtain a reliable estimation of the system state in the presence of sensor attacks.

The main focus of this paper is not the design of the mitigation strategy, but to understand the impact of the detection-mitigation loop on the optimal attack strategy. Thus, in this paper, we do not focus on a specific mitigation approach. Instead, we design a generic framework that admits any mitigation signal. In Section VIII, our simulations are based on a perfect mitigation strategy in which the controller can precisely remove the attack signal, as well as a practical mitigation strategy in which the mitigation signal is a noisy version of the attack signal.

Combining (1), (2), (4) and (5), we obtain the dynamics of the KF estimation error with attack mitigation as

$$\mathbf{e}[t+1] = \mathbf{A}_K \mathbf{e}[t] + \mathbf{W}_K \mathbf{w}[t] - \mathbf{K} \mathbf{a}_m[t+1] - \mathbf{K} \mathbf{v}[t+1], \quad (6)$$

where $\mathbf{a}_m[t+1] = \mathbf{a}[t+1] - i[t+1]\delta[t+1]$, $\mathbf{A}_K = \mathbf{A} - \mathbf{K}\mathbf{C}\mathbf{A}$ and $\mathbf{W}_K = (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{W}$. Further, we have $\mathbb{E}[\mathbf{e}[0]] = \mathbf{0}$ and $\mathbb{E}[\mathbf{e}[0]\mathbf{e}[0]^T] = \mathbf{P}_e = (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{P}_\infty$.

IV. PROBLEM FORMULATION

Under the Kerckhoffs's assumption about the attacker's knowledge, we analyze attack strategies that can mislead the controller into making erroneous control decisions. This is accomplished indirectly by increasing the estimation errors. For a given attack detection threshold η and mitigation strategy $\{\delta[t]\}_{t=1}^T$ over a horizon of T time slots, the optimal attack sequence that maximizes the cumulative sum of KF's expected

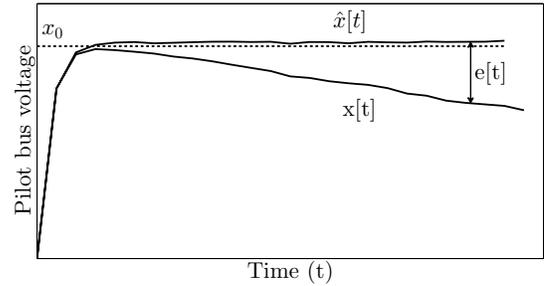


Fig. 3. Attack impact for the voltage control problem.

square of norm of the estimation error over the horizon is given by the following optimization problem:

$$\begin{aligned} \max_{\mathbf{a}[1], \dots, \mathbf{a}[T]} & \sum_{t=1}^T \mathbb{E}[\|\mathbf{e}[t]\|^2] \\ \text{s.t.} & \text{KF error dynamics (6), } \|\mathbf{a}[t]\| \leq a_{\max}, \forall t. \end{aligned} \quad (7)$$

Maximizing the KF estimation error implies that the controller no longer has an accurate estimate of the system state. In systems that use KF for state estimation (such as power grid's voltage control etc.), control input computed based on inaccurate/wrong system state estimates can adversely affect their performance and even result in catastrophic safety incidents. Moreover, the cumulative sum in the objective function implies that the attack has a sustained adverse impact on the system over the entire attack time horizon. We note that similar cumulative metrics have also been widely adopted in control system design to assess the performance of controllers [39].

We now illustrate the relevance of (7) to power grid's voltage control. Fig. 3 shows the impact of an attack that is able to bypass the χ^2 detector (and consequently the controller's mitigation steps) on the pilot bus voltage. In this figure, the dotted line indicates the voltage setpoint, and the solid lines show the evolution of the system state $\mathbf{x}[t]$ and estimate $\hat{\mathbf{x}}[t]$. The gap between the two curves measures

the KF estimation error $\mathbf{e}[t]$. As evident from the figure, if the attacker manages to increase the KF's estimation error using a carefully constructed attack sequence, he can cause a significant deviation of the system state from the desired setpoint. Interestingly, the estimate $\hat{\mathbf{x}}[t]$ is close to the setpoint \mathbf{x}_0 that misleads the controller into believing that the desired setpoint has already been achieved, while the actual pilot bus voltage continues to deviate.

Intuitively, to cause a significant impact, the attack magnitude must be large. But at the same time, it is important that the attack bypasses the controller's detection – otherwise the attack will be mitigated. Thus, the solution of the optimization problem (7) must strike a balance between the attack magnitude and stealthiness. In the following section, we solve the optimization problem (7) using an MDP-based approach.

V. MDP SOLUTION

In this section, we cast the optimization problem (7) to an MDP problem [17] and solve it using value iteration.

Markov Decision Process Model: A state in the MDP corresponds to the KF filter estimation error $\mathbf{e}[t]$ and the actions correspond to the attacker's injection $\mathbf{a}[t]$. Our approach is to map the KF error dynamics (6) to the state transition probabilities of the MDP, and the objective function of (7) to the MDP's long-term expected reward. The mathematical details of the MDP is presented next.

Formally, the MDP is defined by a tuple $(\mathcal{E}, \mathcal{A}, \mathcal{T}, R)$, where $\mathcal{E} \subseteq \mathbb{R}^n$ is the state space of the problem corresponding to the set of all possible $\mathbf{e}[t]$. \mathcal{A} is the action space of the attacker; $\mathcal{T}(\mathbf{e}, \mathbf{a}, \mathbf{e}')$ is the probability of transiting from state \mathbf{e} to \mathbf{e}' (where $\mathbf{e}, \mathbf{e}' \in \mathcal{E}$) under an action $\mathbf{a} \in \mathcal{A}$ of the attacker; formally, $\mathcal{T}(\mathbf{e}, \mathbf{a}, \mathbf{e}') \doteq \mathbb{P}(\mathbf{e}[t+1] = \mathbf{e}' | \mathbf{e}[t] = \mathbf{e}, \mathbf{a}[t+1] = \mathbf{a})$. $R(\mathbf{e}', \mathbf{a}, \mathbf{e})$ is the immediate expected reward for the attacker when it takes an action $\mathbf{a} \in \mathcal{A}$ in state $\mathbf{e} \in \mathcal{E}$.

MDP state transition probabilities: We adopt the following approach. First, we compute the quantity $\mathbb{P}(\mathbf{e}_{\text{lb}} \leq \mathbf{e}[t+1] \leq \mathbf{e}_{\text{ub}} | \mathbf{e}[t] = \mathbf{e}, \mathbf{a}[t+1] = \mathbf{a})$, for any $\mathbf{e}_{\text{lb}}, \mathbf{e}_{\text{ub}} \in \mathbb{R}^n$ and $\mathbf{e}_{\text{lb}} \leq \mathbf{e}_{\text{ub}}$ from (6). Then, we use the fact that for a random variable X , $\mathbb{P}(X = x) \approx \frac{F(-\infty, x+\epsilon) - F(-\infty, x-\epsilon)}{2\epsilon}$, where $F(x_1, x_2) = \mathbb{P}(x_1 \leq X \leq x_2)$ and $\epsilon > 0$ is a small positive quantity. The result is stated in the following lemma:

Lemma V.1: For a given $\mathbf{e}[t] = \mathbf{e}$ and $\mathbf{a}[t+1] = \mathbf{a}$ the quantity $\mathbb{P}(\mathbf{e}_{\text{lb}} \leq \mathbf{e}[t+1] \leq \mathbf{e}_{\text{ub}} | \mathbf{e}[t] = \mathbf{e}, \mathbf{a}[t+1] = \mathbf{a})$ can be computed as the sum of the following terms:

$$\begin{aligned} & \mathbb{P} \left(\begin{bmatrix} 0 \\ \mathbf{e}_{\text{lb}} - \mathbf{y}_2 \end{bmatrix} \leq \mathbf{X} \leq \begin{bmatrix} \eta \\ \mathbf{e}_{\text{ub}} - \mathbf{y}_2 \end{bmatrix} \right) \\ & + \mathbb{P} \left(\begin{bmatrix} \eta \\ \mathbf{e}_{\text{lb}} - \mathbf{y}_2 - \mathbf{K}\delta \end{bmatrix} \leq \mathbf{X} \leq \begin{bmatrix} \infty \\ \mathbf{e}_{\text{ub}} - \mathbf{y}_2 - \mathbf{K}\delta \end{bmatrix} \right). \quad (8) \end{aligned}$$

In (8), $\mathbf{X} \in \mathbb{R}^{n+1}$ is a concatenated variable given by $\mathbf{X} = [\mathbf{Y} (\mathbf{W}_K \mathbf{w}[t] - \mathbf{K}\mathbf{v}[t+1])^T]^T$, $\mathbf{y}_2 = \mathbf{A}_K \mathbf{e} - \mathbf{K}\mathbf{a}$, and δ is the mitigation signal.

The proof of Lemma V.1 is omitted here and presented in Appendix A of [1].

MDP reward: We now map the objective function of (7) to the MDP reward function. Accordingly, the immediate

expected reward of the MDP is given by $R(\mathbf{e}, \mathbf{a}, \mathbf{e}') = \int_{\mathbf{e}' \in \mathcal{E}} \mathcal{T}(\mathbf{e}, \mathbf{a}, \mathbf{e}') \|\mathbf{e}'\|^2$.

MDP policy and state value function: The solution to the MDP corresponds to a policy π , which is a mapping from a state to an action. The state value function of the MDP for a given policy π is defined as $V^\pi(\mathbf{e}) = \mathbb{E}_\pi \left[\sum_{t=1}^T \|\mathbf{e}[t]\|^2 | \mathbf{e}[0] = \mathbf{e} \right]$.

Optimal policy: The optimal policy π^* maximizes the total expected reward, $\pi^* = \arg \max_\pi V^\pi(\mathbf{e}), \forall \mathbf{e} \in \mathcal{E}$, and the optimal value function is defined as $V^*(\mathbf{e}) = V^{\pi^*}(\mathbf{e})$.

Solving the MDP: MDPs can be solved by value/policy iteration methods for discrete systems [17]. However, we study discrete-time CPCS with continuous system states. For instance, the voltages in the voltage control system are continuous variables. Hence, the MDP described does not admit the value iteration method directly. To address this issue, we define a *discretized MDP* by discretizing the state space of the original continuous MDP. The discretized MDP admits a deterministic optimal policy (i.e., a pure strategy) [17], (Theorem 4.4.2), which can be computed using the value iteration method (the convergence of value iteration follows from the fact that the updates of the value iteration algorithm form a contraction mapping [17], Theorem 6.3.1). The optimal policy of the discretized MDP can then be used as a near-optimal solution to the continuous MDP [40]. (Please also refer to our subsequent arguments in bullet point 2.) The details of the discretization procedure are omitted here and can be found in Appendix B of reference [1]. We make some remarks on the MDP formulation:

1. Note that to execute the MDP policy, the attacker requires the knowledge of the Kalman filter estimation error $\mathbf{e}[t]$, which in turn depends on the system state $\mathbf{x}[t]$. The attacker can use a separate Kalman filter to track $\mathbf{x}[t]$, since he can access the original sensor measurements. To simplify our analysis, we assume that the attacker can directly observe $\mathbf{x}(t)$ and $\mathbf{e}(t)$. Under this assumption, the attacker is advantageous. Thus, our analysis is conservative from the perspective of the system defender, which is often acceptable for safety-critical systems. We verify the validity of this assumption using simulations for voltage control in the IEEE-9 bus system and present the result in Section VIII.
2. The optimal cost of the discretized MDP is guaranteed to lie within a bounded distance from the optimal cost of the original MDP [40]. As the discretization is finer, the discretized MDP approaches to the original MDP more closely.
3. Let A and D denote the size of the action and the state space of the discretized MDP. The computational complexity of implementing the value iteration algorithm is $\mathcal{O}(AD)$ [17]. We note that the number of discretization levels of the state space D is an important factor. Moreover, the algorithm requires the value function to be stored for each discretized state, making it challenging to implement in high-dimensional systems. To address this issue, we present a Q-learning algorithm with function approximation in Section VII, which is more suited

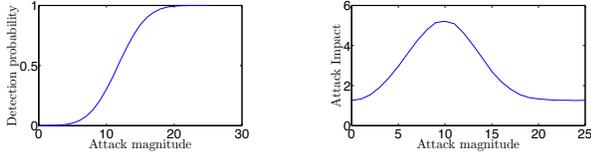


Fig. 4. Attack detection probability and the expected attack impact (immediate expected reward of MDP) for different attack magnitudes.

for application in these systems. We also compare the computational overhead of the two algorithms.

A. Attack Magnitude and Stealthiness

We now illustrate the structure of the MDP solution using a numerical example. In Fig. 4, we plot the attack detection probability and the attack impact computed in terms of the MDP's immediate expected reward for different values of attack magnitude \mathbf{a} . The system parameters are $n = m = 1$, $\mathbf{A} = 1$, $\mathbf{C} = 1$, $\mathbf{Q} = 1$, $\mathbf{R} = 10$, $\eta = 10$ and $\delta = \mathbf{a}$. It can be observed that while the probability of detection is low for an attack of small magnitude, it also has little impact. On the other hand, the probability of detection is high for an attack of large magnitude, and consequently the expected attack impact is also low. The optimal attack lies in between these two quantities. In this example, the optimal attack that maximizes the expected immediate reward has a magnitude of 10, and a detection probability of 0.3. Thus, the MDP solution strikes a balance between attack magnitude and stealthiness, resulting in maximum impact².

VI. COSTS OF FALSE POSITIVES AND MISSED DETECTIONS

In this section, we use the framework established in Section IV and Section V to quantify the costs of false positives and missed detections. We use the cumulative state estimation error (objective function of (7)) as the cost metric. To quantify these costs, we consider an LTI system with an *oracle* attack detector as the reference system. The oracle detector achieves perfect detection capability, that is, it generates no FPs and MDs. The cost of FPs is then the additional cost incurred, compared with the reference system, by wrongly triggered mitigations in the LTI system of Fig. 2 (defined by (1) and (2) with the χ^2 detector and mitigation modules), compared with the reference system. The cost of MDs is the additional cost incurred by unmitigated attacks in the LTI system of Fig. 2.

Cost of FPs: To quantify the cost of FPs, we compute the state estimation error (objective function of (7)) in the following two systems: (i) the LTI system of Fig. 2 under no attacks, i.e., $\mathbf{a}[t] = 0$, $\forall t$; (ii) the reference LTI system with $\mathbf{a}[t] = 0$, $\forall t$. Under setting (i), all the alarms of the χ^2 detector correspond to FPs, which wrongly trigger a mitigation action. Since the mitigation signal is imperfect, it leads to an increase in the estimation error. The difference between

the state estimation errors of the two systems quantifies the performance degradation due to FPs.

Cost of MDs: To quantify the cost of MDs, we compute the state estimation errors in the following two systems: (i) the LTI system of Fig. 2 with optimal attacks (ii) the reference LTI system with optimal attacks. The difference between the state estimation errors of the two systems quantifies the performance degradation due to MD, i.e., the cost of MD. In particular, the optimal attacks derived in Section V characterizes the worst-case performance degradation due to MDs.

It is hard to characterize the state estimation error (objective function of (7)) analytically for a generic LTI system. Nevertheless, for $n = m = 1$, we present a recursive method to compute the state estimation error under any attack sequence $\mathbf{a}[t], \forall t$. We let $\mathcal{B}^t = \{0, 1\}^t$ denote the set of all binary combinations of length t , and $f_{\mathbf{i}_{[1:t]}}(\mathbf{b}) = \mathbb{P}(\mathbf{i}_{[1:t]} = \mathbf{b}), \forall \mathbf{b} \in \mathcal{B}^t$ denote the pdf of $\mathbf{i}_{[1:t]}$. Additionally, we introduce the following notations to denote the conditional random variables $\mathbf{e}_c[t] = \mathbf{e}[t] | \mathbf{i}_{[1:t]}$, $\mathbf{r}_c[t] = \mathbf{r}[t] | \mathbf{i}_{[1:t-1]}$, $\mathbf{e}_c^{(0)}[t+1] = \mathbf{e}[t+1] | \{\mathbf{i}_{[1:t]}, i[t+1] = 0\}$ and $\mathbf{e}_c^{(1)}[t+1] = \mathbf{e}[t+1] | \{\mathbf{i}_{[1:t]}, i[t+1] = 1\}$. The cumulative state estimation error in the LTI system of Fig. 2 can be computed as $\mathbb{E}[\|\mathbf{e}[t]\|^2] = \sum_{\mathbf{b} \in \mathcal{B}^t} f_{\mathbf{i}_{[1:t]}}(\mathbf{b}) \mathbb{E}[\|\mathbf{e}_c[t]\|^2]$, where the terms $\mathbb{E}[\|\mathbf{e}_c[t]\|^2]$ can be evaluated recursively as

$$\mathbb{E}[\|\mathbf{e}_c^{(i)}[t+1]\|^2] = \text{Var}(\mathbf{e}_d^{(i)}[t+1]) + \frac{\int_{a_i}^{b_i} ((\mathbb{E}[\mathbf{e}_d^{(i)}[t+1]])^2 f_{\mathbf{y}}(\mathbf{y}))}{\int_{a_i}^{b_i} f_{\mathbf{y}}(\mathbf{y})}, \quad (9)$$

where $i = 0, 1$. In (9), $\mathbb{E}[\mathbf{e}_d^{(i)}[t+1]]$ and $\text{Var}(\mathbf{e}_d^{(i)}[t+1])$ are the mean and variance of the variable $\mathbf{e}_d^{(i)}[t+1]$. They can be computed using the result presented in Appendix A-Part I. The complete derivation of (9) can be found in Appendix A-Part II.

Using the result above, the cost of FPs can be computed by setting $\mathbf{a}[t] = \mathbf{0}, \forall t$ (i.e., no attacks) and the cost of MDs under any non-zero attack sequence $\mathbf{a}[t], \forall t$ by injecting the appropriate attacks. The worst-case cost of MDs under the optimal attacks (computed as in Section V) can be directly derived from the value function $V(\xi_i)$ at the convergence of the value iteration algorithm.

In Section VIII-B, we present simulation results, based on the derivations in this section, to illustrate the cost of FPs and MDs under different attack detection thresholds and mitigation strategies. We also provide guidelines to tune the attack detection threshold based on this quantification.

VII. ALGORITHM IMPLEMENTATION IN HIGH-DIMENSIONAL SYSTEMS

The value iteration algorithm to solve the MDP (as proposed in Section V) has high computation overhead for high-dimensional systems. Let Ξ denote the discretized version of the original state space \mathcal{E} , where $\Xi = \{\xi_1, \dots, \xi_D\}$, D is the total number discretization levels. Then, the value iteration algorithm loops over each discretized state $\xi_i \in \Xi$, and the value function $V(\xi_i)$ must be stored for each discretized state $\xi_i \in \Xi$ (refer to Algorithm 1 in reference [1] for details of the

²Strictly speaking, MDP solution maximizes the long term expected reward. For the ease of illustration, in this example we only considered the immediate expected reward.

value iteration algorithm). For an n -dimensional system, if each dimension is discretized into d states, the total number of discretized states is $D = d^n$. Thus, it becomes challenging to run the value iteration algorithm. In this section, we propose to use the *Q-learning with function approximation* (QLFA) method [41] to extend the MDP solution to high-dimensional systems.

We first introduce the concept of Q-learning. It is a model-free method for computing the optimal MDP policy that maximizes the time-average reward. The basic idea of Q-learning is to perform *trajectory-based sampling*, i.e., run trajectories of the Kalman filter estimation error $\langle \mathbf{e}[t], \mathbf{a}[t], R[t], \mathbf{e}[t+1], \mathbf{a}[t+1], R[t+1], \dots \rangle$, and perform Bellman backups only on the states visited. Thus, the Q-learning algorithm does not loop over each state $\xi_i \in \Xi$, in contrast to the value iteration algorithm. The Q-learning algorithm computes the action-value function for each state and action pair $Q(\xi_i, \mathbf{a})$. In particular, for $\xi_i[t] = \xi_i, \mathbf{a}[t] = \mathbf{a}, \mathbf{e}[t+1] = \xi'_i$, and $R[t+1] = R$, the Q-values are updated as

$$Q(\xi_i, \mathbf{a}) \leftarrow (1-\alpha)Q(\xi_i, \mathbf{a}) + \alpha \left(R + \gamma \max_{\mathbf{a}'} Q(\xi'_i, \mathbf{a}') \right), \quad (10)$$

where $\alpha > 0$ is the learning rate. The action $\mathbf{a}[t] = \mathbf{a}$ is chosen according to the ϵ -greedy method [41]. The Q-learning algorithm can converge for a finite number of state-action pairs under certain conditions on the learning rate, discount factor, and the reward function [41].

The Q-learning algorithm still has the issue that it stores the Q-values for each state-action pair, which will be challenging in high-dimensional systems. To address this issue, we propose to combine the Q-learning with the *function approximation* method [42]. The key idea of this approach is to approximate the Q-values using parametric functions instead of storing them individually for each state-action pair. The number of parameters to be stored and updated is much less than the number of states. As a result, this method is more suited for application in high-dimensional systems. In particular, we focus on Q-learning with linear function approximation (QLFA), in which the Q-values are approximated as a linear function of a parameter vector $\boldsymbol{\theta}(\mathbf{a}), \forall \mathbf{a}$ as

$$Q(\xi_i, \mathbf{a}) = \phi(\xi_i)^T \boldsymbol{\theta}(\mathbf{a}), \quad \forall \xi_i \in \Xi, \mathbf{a} \in \mathcal{A}, \quad (11)$$

where $\phi(\xi_i)$ is the *feature vector* corresponding to every state ξ_i . The feature vector is a lower-dimensional representation of the state, where the feature function $\phi : \mathcal{E} \rightarrow \mathbb{R}^K$ maps each state-action pair to a feature vector, K being the dimension of the feature vector. The feature function ϕ can be constructed for each state using various *basis functions*. For example, one of the simplest basis functions is the *fixed sparse representation* (FSR) with binary encoding. Under FSR, we first discretize the n -dimensional state space with uniform grid cells of d levels each. Thus, the state space Ξ has d^n states represented by the tuple $(i, j, \dots) \in \Xi$. The feature vector corresponding to each state $(i, j, \dots) \in \Xi$ is then constructed

as

$$\phi((i, j, \dots)) = [0, \dots, \underbrace{1}_{i^{\text{th}} \text{ position}}, \dots, 0, 0, \dots, \underbrace{1}_{j^{\text{th}} \text{ position}}, \dots, 0, \dots]^T. \quad (12)$$

$\underbrace{\hspace{10em}}_{d \text{ elements}} \quad \underbrace{\hspace{10em}}_{d \text{ elements}}$

Note that $\phi((i, j, \dots))$ is a nd -dimensional vector (thus, in this case, $K = nd$).

Other choices of basis functions to construct the feature vector include the use of Radial Basis Functions (RBFs) and Incremental Feature Dependency Discovery (iFDD) [42]. The choice of basis function offers a trade-off between accuracy (of estimating the correct value of the MDP's value function) and learning speed.

Under the function approximation method, finding the optimal policy involves updating the parameter vector $\boldsymbol{\theta}(\mathbf{a})$ (rather than the Q-values). Thus, the total number of elements to be stored is ndA (as opposed to $DA = d^n A$ for the direct Q-learning algorithm). We specify the QLFA algorithm in the following.

Algorithm 1: Q-Learning with function approximation

Data: MDP, α, γ

Result: Policy π

$\boldsymbol{\theta}(\mathbf{a}) \leftarrow$ Initialize arbitrarily $\forall \mathbf{a}$;

$\langle \xi, \mathbf{a} \rangle \leftarrow \langle \xi_0, \pi^\epsilon(\xi_0) \rangle$. **while** *time left* **do**

Take action a , observe reward R and next state ξ'

$$Q^+(\xi_i, \mathbf{a}) \leftarrow R + \gamma \max_{\mathbf{a}'} Q(\xi'_i, \mathbf{a}') \quad \mathcal{O}(ndA^2)$$

$$\delta \leftarrow Q^+(\xi_i) - Q(\xi_i)$$

$$\boldsymbol{\theta}(\mathbf{a}) \leftarrow \boldsymbol{\theta}(\mathbf{a}) + \alpha \delta \phi(\xi_i) \quad \mathcal{O}(ndA)$$

$$\langle \xi', \mathbf{a} \rangle \leftarrow \langle \xi', \pi^\epsilon(\xi') \rangle \quad \mathcal{O}(ndA^2)$$

end

Return π greedy with respect to Q

Computational complexity of QLFA algorithm: The computational complexity of main execution steps of the QLFA algorithm is indicated in Algorithm 2. We note that computing $Q(\xi_i, \mathbf{a})$ has a complexity of $\mathcal{O}(ndA)$ (since it involves multiplication of two vectors of dimension ndA as in eq. (11)). Thus, computing the policy $\max_a Q(\xi_i, a)$ has a complexity of $\mathcal{O}(ndA^2)$. We observe that the complexity of the algorithm is linear in the system dimension n , as opposed to the exponential complexity of the value iteration and the original Q-learning algorithm. Moreover, the number of parameters to be stored and updated is ndA . Thus, The QLFA algorithm is feasible even for high-dimensional systems.

In Section VIII, we conduct simulations to compare the performance of attack sequences generated by the QLFA algorithm with other attack sequences in high-dimensional systems.

A. Q-learning with Non-Linear Function Approximation

Although the QLFA algorithm is computationally lightweight, its performance often requires a careful design of the feature set, which is non-trivial and challenging [42]. Without the right set of features, the linear function

approximators may not accurately represent the true value function. Alternately, one can use non-linear function approximators (NLFA) such as a neural network. The NLFA offer a richer class of function approximators that can map directly from the states to the value function without the need to explicitly specify the feature set. In particular, we use the Q-learning with non-linear function approximation (Q-NLFA) algorithm [43] to solve the MDP (7). Simulation results for the Q-NLFA algorithm are presented in Section VIII-E, where we show that it outperforms the QLFA algorithm in high-dimensional systems.

Finally, note that in this section, we have used the *value function approximation* method (i.e., we approximated the value function using linear/non-linear approximators). An alternative function approximation approach in high-dimensional/continuous systems is the *policy gradient* method, which approximates the policy directly instead of the value function [44]. While policy gradient methods are well-suited for high-dimensional/continuous systems, they are known to exhibit high variance in estimating the gradient, which may adversely affect the learning.

VIII. SIMULATION RESULTS

We perform simulations on the voltage control system using PowerWorld, which is a high-fidelity power system simulator widely used in the industry [45]. All the simulations are performed based on the IEEE 9-bus system shown in Fig. 1, in which buses 1, 2, and 3 are the generator buses, buses 5, 6 and 8 are the load buses (we use the default load values), and 5, 7, and 9 are the pilot buses. The control matrix \mathbf{B} is estimated using linear regression on the data traces of $\mathbf{x}[t+1] - \mathbf{x}[t]$ and $\mathbf{u}[t]$ obtained in a PowerWorld simulation. Specifically, we applied a series of control inputs (i.e., changed the voltage of the generator bus) and observed the change in voltage of the pilot bus in Powerworld simulation, and used linear regression between the two data series to obtain \mathbf{B} . Further, the process noise and observation noise are assumed to be Gaussian with standard deviation of 0.01 and 0.02 pu respectively. The choice is consistent with real-world sensor measurement noise from PMUs [46].

First, we verify the accuracy of the LTI model in approximating the real-world voltage control system by examining the voltage at pilot bus 5. In our simulations, the voltage controller aims to adjust the voltage of this bus from an initial voltage of 1 pu to a setpoint (\mathbf{x}_0) of 0.835 pu (base voltage of 230 kV) by applying the control by applying the control $\mathbf{u}[t] = \mathbf{B}^{-1}(\mathbf{x}_0 - \hat{\mathbf{x}}[t])$ in both Powerworld and the LTI model. Fig. 5 plots the bus voltage from $t = 1$ to $t = 30$ obtained from the PowerWorld simulations, as well as the voltage values obtained from the LTI model. To average the effect of random measurement noise, we repeat the experiment 100 times, and take the mean value. The two curves match well in this figure, thus verifying the accuracy of the proposed LTI model.

We also include a simulation result to show the validity of the approximation that the attacker can accurately track the KF's estimation error (see bullet point 1 in Section V). We consider the IEEE-9 bus system with similar settings as

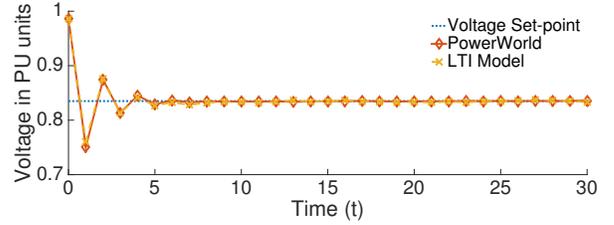


Fig. 5. Comparison between PowerWorld and LTI model.

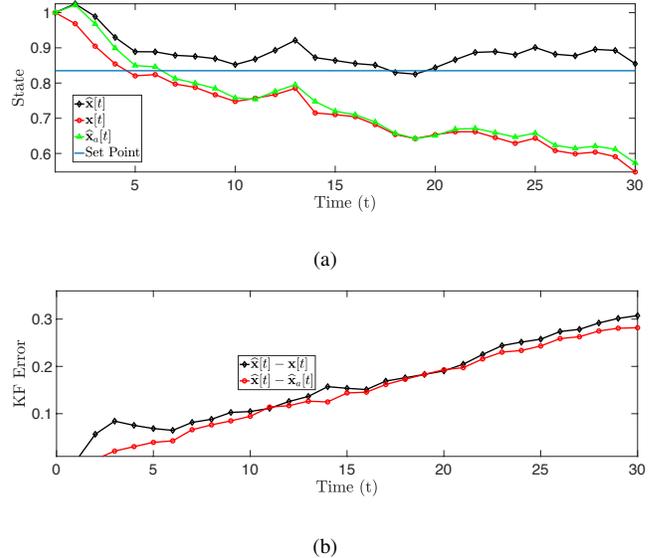


Fig. 6. IEEE-9 bus system under ramp attack. (a) Evolution of system state and the estimate. (b) Evolution of the KF estimation error

above and under a ramp attack sequence. We set the slope of the ramp to 0.01 pu. For simplicity, we do not consider attack detection (since our objective is to show the attacker's estimate of the system state under an attack sequence that bypasses detection). Denote the defender's and the attacker's estimates of the system state by $\hat{\mathbf{x}}[t]$ and $\tilde{\mathbf{x}}_a[t]$ respectively, which are computed as

$$\begin{aligned}\hat{\mathbf{x}}[t+1] &= \mathbf{A}\hat{\mathbf{x}}[t] + \mathbf{B}\mathbf{u}[t] + \mathbf{K}(\mathbf{y}_f[t+1] - \mathbf{C}(\mathbf{A}\hat{\mathbf{x}}[t] + \mathbf{B}\mathbf{u}[t])), \\ \tilde{\mathbf{x}}_a[t+1] &= \mathbf{A}\tilde{\mathbf{x}}_a[t] + \mathbf{B}\mathbf{u}[t] + \mathbf{K}(\mathbf{y}[t+1] - \mathbf{C}(\mathbf{A}\hat{\mathbf{x}}[t] + \mathbf{B}\mathbf{u}[t])).\end{aligned}$$

Here $\mathbf{y}_f[t] = \mathbf{y}[t] - i[t]\delta[t]$ and $\mathbf{u}[t] = \alpha(\mathbf{x}_0 - \hat{\mathbf{x}}[t])$. Note that the attacker's estimate is computed using $\mathbf{y}[t+1]$ (i.e., the true sensor measurements).

The result of the simulation is plotted in Fig. 6 (a). It can be seen that while the defender (i.e., the system operator) has an erroneous estimate of the system state due to the FDI attack, the attacker can track the system state fairly accurately based on the true sensor measurements. We also plot the KF estimation error (i.e., $\hat{\mathbf{x}}[t] - \mathbf{x}[t]$), and the attacker's estimate of the KF error (i.e., $\hat{\mathbf{x}}[t] - \tilde{\mathbf{x}}_a[t]$) in Fig. 6 (b), which shows a close match between the two curves. Hence, in our analysis, we assume that the attacker can track the Kalman filter's estimation error $\mathbf{e}[t]$ accurately.

A. Attacks Against Voltage Control Using MDP and Value Iteration Algorithm

Next, we simulate the impact of the proposed attacks on the voltage control system while considering the χ^2 attack detection. We assume that the attacker has access to the voltage sensor of bus 5, and injects false measurements to mislead the controller. We compute the optimal attack sequence based on the LTI model using the value iteration method implemented in MATLAB. Recall that in this example, the state of the MDP is the voltage of bus 5 and the attack sequence is the FDI attack on the voltage sensor measurement of bus 5. We restrict $\mathbf{a}[t]$ between $0 - 0.2$ pu discretized with an interval of 0.02 pu. To evaluate the attack impact, we run Monte Carlo simulations using the PowerWorld simulator by injecting the derived optimal attack into the voltage measurements, and implementing the control based on the corresponding state estimate. Fig. 7 (top figure) shows the pilot bus voltage (bus 5) for different attack sequences with $\eta = 5$ and perfect attack mitigation. It can be observed that the pilot bus voltage deviates from the setpoint of 0.835 pu, and the largest voltage deviation is seen under the optimal attack. In particular, over an attack duration of 30 time slots, we observe that bus 5 voltage deviates to 0.65 pu under the optimal attack, which is about 0.2 pu from the setpoint.

Fig. 7 (bottom figure) shows the attack detection probability under these attacks at different time instants. We also plot the optimal policy computed by the value iteration algorithm (Algorithm 1) in Fig. 8a, and the optimal attack sequence for three Monte Carlo instantiations in Fig. 8b. We observe that the attack detection probability for a naive attack sequence such as the ramp attack increases with time, which results in nullifying its impact due to attack mitigation. However, the optimal attack is crafted in a way such that the detection probability decreases over time. Consequently, the optimal attack causes a significant deviation of the pilot bus voltage from its setpoint.

Algorithm Performance under Non-Gaussian Noises: We also perform simulations to examine the algorithm performance under non-Gaussian noise settings. Recent work based on real PMU noise data shows that its statistical distribution can be modeled by non-Gaussian distributions such as the Student-t or Laplace distributions [28]. Accordingly, we examine the performance of the policy derived using the MDP algorithm in a power system whose measurement noises follow a non-Gaussian distribution. Specifically, we derive the policy using the MDP approach (which assumes Gaussian noise). However, at the policy evaluation stage, we evaluate using two non-Gaussian noise distributions, i.e., Logistic and Student-t distributions and compare the time average cost function (time average Kalman filter estimation error). For fair comparison, we set the noise variance to be the same as that of the Gaussian case (0.02 pu). The results are shown in Fig. 9 (a), in which we plot the time average cost function. It can be seen that the performance of the MDP algorithm with non-Gaussian noise is nearly identical to the Gaussian noise setting. We also plot the evolution of the pilot bus voltage in the presence of attack based on the MDP policy in Fig. 9 (b). The performance degradation is nearly identical in all the three cases. Hence,

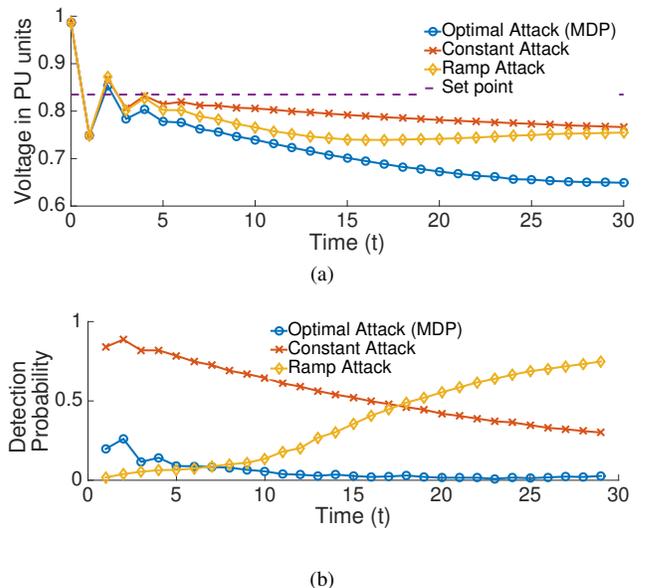


Fig. 7. Attack against voltage control system. (a) Pilot bus voltage (Bus 5) under different attack sequences. (b) Attack Detection probability for different attacks.

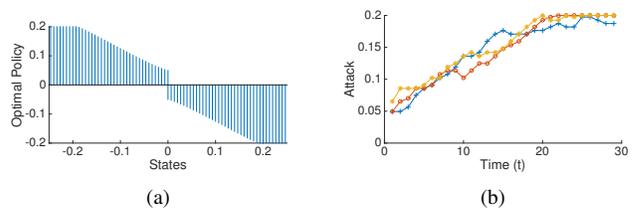


Fig. 8. (a) Optimal policy for different system states computed by value iteration. (b) Optimal attack sequence for 3 Monte Carlo simulation instantiations.

we conclude that the MDP algorithm is nearly optimal for real-world PMU sensor measurement noises.

B. Quantifying Cost of False Positives and Missed Detections

In this subsection, we perform simulations to quantify the cost of FPs and MDs following the approach in Section VI. We consider a general LTI model described by (1) and (2) with $n = 1$, $\mathbf{A} = 1$, $\mathbf{C} = 1$, $\mathbf{Q} = 1$, $\mathbf{R} = 10$ and adopt a *practical attack mitigation approach* under which the attack mitigation signal is given by $\delta_d[t] = \mathbf{a}[t] + \mathbf{b}[t]$, where $\mathbf{b}[t]$ is the error component in the mitigation signal. We generate $\mathbf{b}[t]$ as a Gaussian distributed random variable with a standard deviation of σ_{mit} units. Fig. 10 shows the cost of FPs and MDs for different detection thresholds η and standard deviations of the attack mitigation signal σ_{mit} .

From these plots, we observe that as the attack detection threshold η is increased, the cost of FP decreases and the cost of MD increases. This result is intuitive – a low detection threshold detects most attacks but also leads to a large number of FPs. Thus, the wrongly triggered mitigations will result in a high FP cost. On the other hand, a high detection threshold yields a low number of FPs, but also increases the number of MDs. The figures show a basic tradeoff between FPs and MDs, quantified in terms of the cost function.

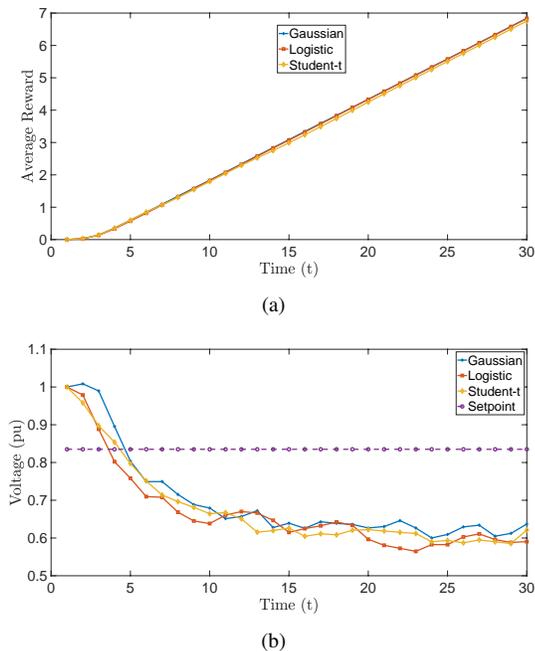


Fig. 9. (a) Average reward of the MDP policy, (b) pilot bus voltages under non-Gaussian noise settings.

We also observe that these costs depend on the accuracy of the attack mitigation signal. For instance, when the accuracy is high (e.g., $\sigma_{\text{mit}} = 0, 5$), the cost of FP is very low, even for a low detection threshold. Thus, in this scenario, the system operator can choose a low detection threshold and obtain good overall system performance. However, when the accuracy of the mitigation signal is low (e.g., $\sigma_{\text{mit}} = 15$), the cost of FP is high for a low detection threshold. For instance, in Fig. 10(d), the cost of FP for $\eta = 0$ is greater than the cost of MD for $\eta = 5$. In this scenario, the system operator must choose a high detection threshold to obtain an acceptable level of system performance. Thus, our result helps the system operator select an appropriate threshold to balance the costs of FP and MD, depending on the accuracy of the mitigation signal.

Lastly, we note that for $\sigma_{\text{mit}} = 0$ (i.e., perfect mitigation), the cost of FP is zero for all detection thresholds. Under perfect mitigation, even if an FP event occurs, the controller can accurately estimate that the attack magnitude is zero (i.e., no attack). Thus, in this specific case, wrongly triggered mitigations do not increase the cost of FP. We also note that for $\eta = 0$, there are no MDs. Hence, the cost of MD in this case is nearly zero.

C. Comparison of Value Iteration, Q-Learning and QLFA Algorithms

For the LTI system described in previous subsection, we compare the performance of value iteration, Q-learning (eq. (10)) and QLFA algorithms in a 1-dimensional system. For QLFA, we use FSR with binary encoding as the basis function. We first compute the policy $\pi(\mathbf{e})$ using each of the three algorithms and then evaluate the cost function using the corresponding policies. We repeat the simulations for different

numbers of discretization levels $d = \{21, 41, 81\}$. The results are shown in Fig. 11a, which show that the performance of the original Q-learning algorithm and the QLFA algorithm is close to that of the value iteration algorithm. We also plot the policies of the three algorithms in Fig. 11b, which are similar.

D. Simulations for High-Dimensional Systems Using Q-Learning with Function Approximation

We report simulations in high-dimensional systems using the QLFA (with FSR with binary encoding as the basis function) and Q-NLFA algorithms proposed in Section VII. We use two bus systems (i) a moderately-sized IEEE-39 bus system which has 10 generator buses (bus 30 – 39) and 10 pilot buses and (ii) a large bus system, i.e., the IEEE-118 bus system which has 30 pilot buses (we only assign the non-generator buses as pilot buses). The state of the system is the pilot bus voltages and the actions are the FDI attack injected at the pilot bus voltage sensor measurements. We estimate the control matrix \mathbf{B} for both the bus systems using linear regression on data traces obtained in a PowerWorld simulation.

First, we show the average reward of the QLFA and Q-NLFA algorithms as a function of the training time in Fig. 12 for both the bus systems. To derive the policy for the QLFA algorithm, we compute the weight vector $(\theta(\mathbf{a}))$ for different training times in a MATLAB simulation. To derive the policy for the Q-NLFA algorithm, we use the TensorFlow library [47]. In particular, we construct a neural network with 20 nodes per layer and 5 hidden layers and employ the rectified linear unit (ReLU) activation function. We set a batch size of 200 for deep Q-learning’s *experience replay* step [43].

After obtaining the policy in each case, we run test trajectories with the corresponding policies, and obtain the average reward (cumulative Kalman filter estimation error) over 50 time slots. It can be observed from Fig. 12 that the average rewards of the QLFA and the Q-NLFA algorithms increase with the training time, implying that the algorithms learn policies that yield a high reward. Moreover, we observe that for the IEEE-39 bus system, the Q-NLFA algorithm converges to a policy that yields high reward much faster than the QLFA algorithm. For a larger system such as the IEEE-118 bus, the Q-NLFA algorithm significantly outperforms the QLFA algorithm, and the QLFA algorithm is suboptimal. Thus, we conclude that Q-NLFA algorithm should be used for high-dimensional systems (with more than 10 pilot buses).

Next, we use the policy obtained from the Q-NLFA algorithm as the attack vector in voltage control loop of the IEEE-39 bus system in a PowerWorld simulation. In these simulations, the controller adjusts the voltage of the pilot buses from an initial voltage of 0.3 pu to 0.835 pu by applying the control $\mathbf{u}[t] = \mathbf{B}^{-1}(\mathbf{x}_0 - \hat{\mathbf{x}}[t])$. The attacker injects the attack vector on the sensor measurements of the pilot buses. The resulting pilot bus voltages and their estimates are plotted in Fig. 13. It can be observed that while the voltage estimates are close to the set-point, the actual pilot bus voltages deviate significantly from the set-point, thus showing the efficacy of the derived attacks.

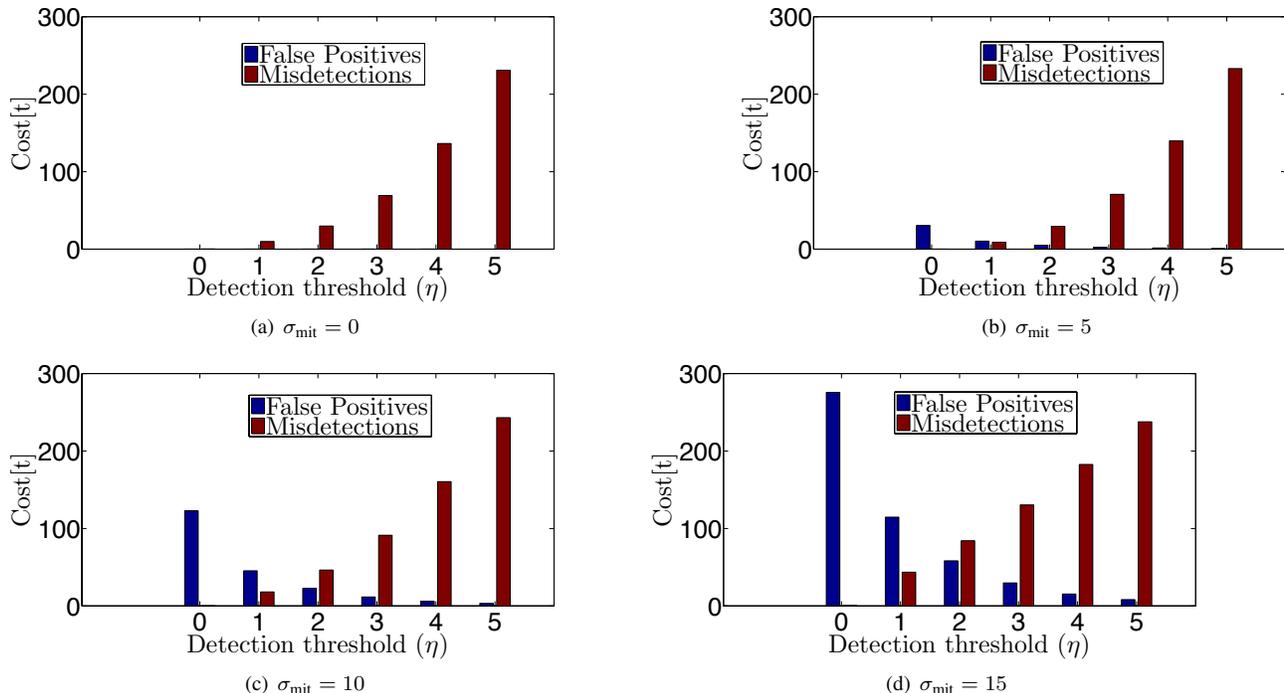
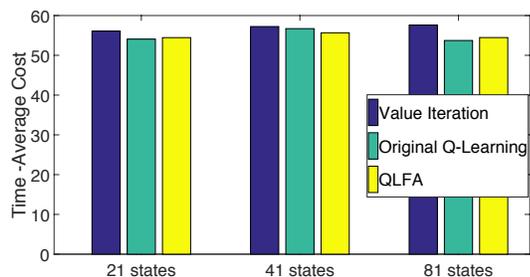
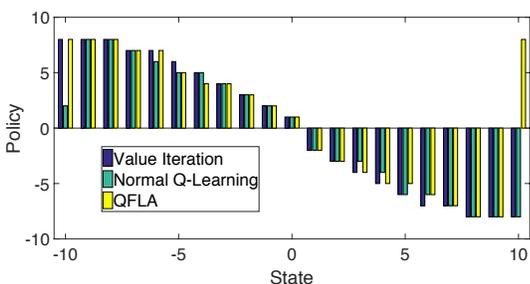


Fig. 10. Cost of FPs and MDs for different attack detection thresholds and standard deviation of the attack mitigation signal σ_{mit} .

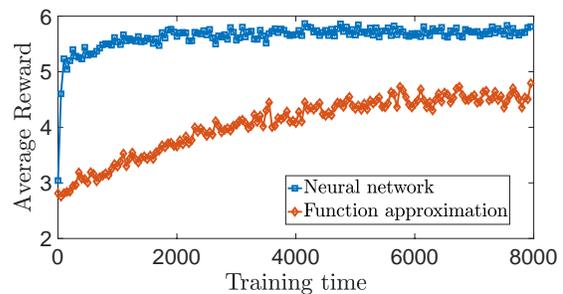


(a)

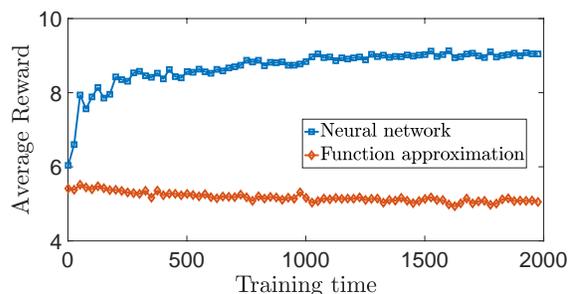


(b)

Fig. 11. (a) Time average reward under policies from value iteration, naive QL and QLFA algorithms in 1-D system. (b) Policies from the respective algorithms for $d = 21$ states.



(a)



(b)

Fig. 12. Comparison of QLFA and Q-NLFA algorithms for different training times (a) IEEE-39 bus system (with 10 pilot buses), (b) IEEE-118 bus system (with 30 pilot buses).

IX. CONCLUSIONS

In this paper, we studied the performance of a CPCS with attack detection and reactive attack mitigation. We derived the optimal attack sequence that maximizes the state estimation error over the attack time horizon using an MDP framework. Our results show that an arbitrarily constructed attack sequence will have little impact on the system since it will be detected,

hence mitigated. The optimal attack sequence must be crafted to strike a balance between the stealthiness and the attack magnitude. Our results are useful for the system operator to assess the limit of attack impact and compare different attack detection and mitigation strategies. We also quantified the impact of FP and MD on the state estimation error, which

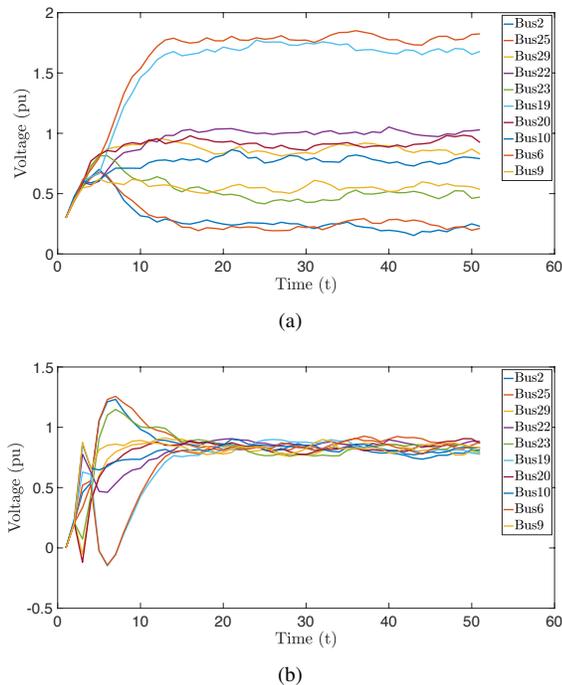


Fig. 13. Attack against voltage control system for the IEEE-39 bus. (a) Pilot bus voltages under QLFA attacks. (b) Estimates of the pilot bus voltages.

helps select the right attack detection threshold depending on the accuracy of the attack mitigation signal. We apply our results to the voltage control in a power system.

REFERENCES

- [1] S. Lakshminarayana, T. Z. Teng, D. K. Y. Yau, and R. Tan, "Optimal attack against cyber-physical control systems with reactive attack mitigation," in *Proc. ACM International Conference on Future Energy Systems (e-Energy)*, 2017, pp. 179–190.
- [2] "Confirmation of a coordinated attack on the Ukrainian power grid," 2016, <http://bit.ly/1OmxfnG>.
- [3] Y. Mo and B. Sinopoli, "On the performance degradation of cyber-physical systems under stealthy integrity attacks," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2618–2624, Sep. 2016.
- [4] C. Kwon, W. Liu, and I. Hwang, "Security analysis for cyber-physical systems against stealthy deception attacks," in *Proc. American Control Conference*, June 2013, pp. 3344–3349.
- [5] C. Z. Bai and V. Gupta, "On Kalman filtering in the presence of a compromised sensor: Fundamental performance bounds," in *Proc. American Control Conference*, June 2014, pp. 3029–3034.
- [6] C. Barreto, A. A. Cárdenas, and N. Quijano, "Controllability of dynamical systems: Threat models and reactive security," in *Proc. International Conference on Decision and Game Theory for Security*, 2013, pp. 45–64.
- [7] Q. Zhu and T. Basar, "Game-theoretic methods for robustness, security, and resilience of cyber-physical control systems: Games-in-games principle for optimal cross-layer resilient control systems," *IEEE Control Syst. Mag.*, vol. 35, no. 1, pp. 46–65, Feb 2015.
- [8] J. Paul and J. Leost, "Improvements of the secondary voltage control in france," in *Power Systems and Power Plant Control*, ser. IFAC Symposia Series, 1987, pp. 83 – 88.
- [9] M. D. Ilic, X. Liu, G. Leung, M. Athans, C. Vialas, and P. Pruvot, "Improved secondary and new tertiary voltage control," *IEEE Trans. Power Syst.*, vol. 10, no. 4, pp. 1851 – 1862, Nov 1995.
- [10] P. Kundur, *Power System Stability and Control*. McGraw-Hill, New York, US, 1994.
- [11] A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*. A Wiley-Interscience, 1996.
- [12] A. Monticelli, "Electric power system state estimation," *Proc. IEEE*, vol. 88, no. 2, Feb 2000.
- [13] F. Pasqualetti, F. Dörfler, and F. Bullo, "Cyber-physical attacks in power networks: Models, fundamental limitations and monitor design," in *IEEE Conference on Decision and Control and European Control Conference*, Dec 2011, pp. 2195–2201.
- [14] C. Zhao, U. Topcu, N. Li, and S. Low, "Design and stability of load-side primary frequency control in power systems," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1177–1189, May 2014.
- [15] S. Amini, F. Pasqualetti, and H. Mohsenian-Rad, "Dynamic load altering attacks against power system stability: Attack models and protection schemes," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2862–2872, July 2018.
- [16] R. Mehra and J. Peschon, "An innovations approach to fault detection and diagnosis in dynamic systems," *Automatica*, vol. 7, no. 5, pp. 637 – 640, 1971.
- [17] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1994.
- [18] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov 2013.
- [19] L. Xie, Y. Mo, and B. Sinopoli, "Integrity data attacks in power market operations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 659–666, Dec 2011.
- [20] Y. Yuan, Z. Li, and K. Ren, "Modeling load redistribution attacks in power systems," *IEEE Trans. Smart Grid*, vol. 2, no. 2, 2011.
- [21] R. Tan, H. H. Nguyen, E. Y. S. Foo, X. Dong, D. K. Y. Yau, Z. Kalbarczyk, R. K. Iyer, and H. B. Gooi, "Optimal false data injection attack against automatic generation control in power grids," in *ACM/IEEE ICCPS*, Apr. 2016, pp. 1–10.
- [22] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye, "Detecting false data injection attacks on DC state estimation," in *Proc. Workshop on Secure Control Systems (SCS)*, Apr. 2010. [Online]. Available: <https://tinyurl.com/mh9g99c>
- [23] G. Dan and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," in *Proc. IEEE International Conference on Smart Grid Communications*, Oct 2010, pp. 214–219.
- [24] T. T. Kim and H. V. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Trans. on Smart Grid*, vol. 2, no. 2, pp. 326–333, June 2011.
- [25] M. Esmalifalak, G. Shi, Z. Han, and L. Song, "Bad data injection attack and defense in electricity market using game theory study," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 160–169, March 2013.
- [26] A. Sanjab and W. Saad, "Data injection attacks on smart grids with multiple adversaries: A game-theoretic perspective," *IEEE Trans. Smart Grid*, vol. 7, no. 4, pp. 2038–2049, July 2016.
- [27] C. Y. T. Ma, D. K. Y. Yau, X. Lou, and N. S. V. Rao, "Markov game analysis for attack-defense of power networks under possible misinformation," *IEEE Trans. Power Syst.*, pp. 1676–1686, May 2013.
- [28] S. Wang, J. Zhao, Z. Huang, and R. Diao, "Assessing Gaussian assumption of PMU measurement error using field data," *IEEE Trans. Power Delivery*, vol. 33, no. 6, pp. 3233–3236, Dec. 2018.
- [29] M. B. D. C. Filho and J. C. S. de Souza, "Forecasting-aided state estimation—part i: Panorama," *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1667–1677, Nov 2009.
- [30] C. Carquex, C. Rosenberg, and K. Bhattacharya, "State estimation in power distribution systems based on ensemble kalman filtering," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6600–6610, Nov 2018.
- [31] "Shodan," <https://www.shodan.io/>.
- [32] "The dragonfly attack," 2014, <https://bit.ly/1RzGx2P>.
- [33] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *Conf. IEEE Industrial Electronics Society*, 2011.
- [34] L. A. Garcia, F. Brasser, M. Cintuglu, A.-R. Sadeghi, O. Mohammed, and S. A. Zonouz, "Hey, my malware knows physics! attacking ples with physical model aware rootkit," in *Proc. NDSS*, 2017.
- [35] A. A. Cárdenas, S. Amin, Z. Lin, Y. Huang, C. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," in *Proc. ACM Asia Conference on Computer and Communications Security*, 2011, pp. 355–366.
- [36] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 580–591, March 2014.
- [37] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1454–1467, June 2014.
- [38] S. Mishra, Y. Shoukry, N. Karamchandani, S. N. Diggavi, and P. Tabuada, "Secure state estimation against sensor attacks in the

presence of noise," *IEEE Trans. on Control of Network Systems*, vol. 4, no. 1, pp. 49–59, Mar. 2017.

- [39] T. Abdelzaher, Y. Diao, J. L. Hellerstein, C. Lu, and X. Zhu, "Introduction to control theory and its application to computing systems," in *Performance Modeling and Engineering*. Springer, 2008, pp. 185–215.
- [40] C. S. Chow and J. N. Tsitsiklis, "An optimal one-way multigrid algorithm for discrete-time stochastic control," *IEEE Trans. Autom. Control*, vol. 36, no. 8, pp. 898–914, Aug 1991.
- [41] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998.
- [42] N. Roy and J. How, *A tutorial on linear function approximators for dynamic programming and reinforcement learning*. Now Publishers Inc., 2013.
- [43] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," 2013. [Online]. Available: <http://arxiv.org/abs/1312.5602>
- [44] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. NIPS*, 1999, pp. 1057–1063.
- [45] "PowerWorld," 2017, <http://www.powerworld.com>.
- [46] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in pmu data," in *IEEE Power and Energy Society General Meeting (PESGM)*, July 2016, pp. 1–5.
- [47] "Tensorflow," <https://www.tensorflow.org/>.

APPENDIX A: COMPUTATION OF THE CUMULATIVE STATE ESTIMATION ERROR

Appendix A - Part I

We first state the result for the computation of $\mathbb{E}[\mathbf{e}_d^{(i)}[t+1]]$ and $\text{Var}(\mathbf{e}_d^{(i)}[t+1])$.

The mean of $\mathbf{e}_d^{(i)}[t+1]$ can be computed as

$$\mathbb{E}[\mathbf{e}_d^{(i)}[t+1]] = \bar{\mathbf{x}}^{(i)} + \Sigma_{\mathbf{xy}} \Sigma_{\mathbf{yy}}^{-1} (\mathbf{y} - \bar{\mathbf{y}}),$$

where, $\bar{\mathbf{x}}^{(i)} = \mathbf{A}_K \mathbb{E}[\mathbf{e}_c[t]] - \mathbf{K}(\mathbf{a}[t+1] - \mathbb{1}_{\{i=1\}} \bar{\boldsymbol{\delta}}[t+1])$, $\bar{\mathbf{y}} = \mathbf{C} \mathbf{A} \mathbb{E}[\mathbf{e}_c[t]] + \mathbf{a}[t+1]$. The variance of $\mathbf{e}_d^{(i)}[t+1]$ can be computed as

$$\text{Var}(\mathbf{e}_d^{(i)}[t+1]) = \Sigma_{\mathbf{xx}}^{(i)} - \Sigma_{\mathbf{xy}} \Sigma_{\mathbf{yy}}^{-1} \Sigma_{\mathbf{yx}},$$

where,

$$\begin{aligned} \Sigma_{\mathbf{yy}} &= \mathbf{C} \mathbf{A} \text{Var}(\mathbf{e}_c[t]) \mathbf{A}^T \mathbf{C}^T + \mathbf{C} \mathbf{Q} \mathbf{C}^T + \mathbf{R}, \\ \Sigma_{\mathbf{xy}} &= \mathbf{A}_K \text{Var}(\mathbf{e}_c[t]) \mathbf{A}^T \mathbf{C}^T + \mathbf{W}_K \mathbf{Q} \mathbf{C}^T - \mathbf{K} \mathbf{R}^T, \\ \Sigma_{\mathbf{yx}} &= \mathbf{C} \mathbf{A} \text{Var}(\mathbf{e}_c[t]) \mathbf{A}_K^T + \mathbf{C} \mathbf{Q} \mathbf{W}_K^T - \mathbf{R} \mathbf{K}^T, \\ \Sigma_{\mathbf{xx}}^{(i)} &= \mathbf{A}_K \text{Var}(\mathbf{e}_c[t]) \mathbf{A}_K^T + \mathbf{W}_K \mathbf{Q} \mathbf{W}_K^T + \mathbf{K} \mathbf{R} \mathbf{K}^T + \mathbb{1}_{\{i=1\}} \mathbf{K} \mathbf{B} \mathbf{K}^T, \end{aligned}$$

$\mathbb{1}_x = 1$ if x is true, or 0 if x is false, $\mathbf{B} = \mathbb{E}[(\boldsymbol{\delta}[t+1] - \mathbb{E}[\boldsymbol{\delta}[t+1]]) (\boldsymbol{\delta}[t+1] - \mathbb{E}[\boldsymbol{\delta}[t+1]])^T]$ and $f_{\mathbf{y}}(y)$ is the Gaussian pdf with mean $\bar{\mathbf{y}}$ and variance given by $\Sigma_{\mathbf{yy}}$. The integration limits a_i and b_i are $[-\sqrt{\eta} \mathbf{P}_r, \sqrt{\eta} \mathbf{P}_r]$, if $i = 0$, and $(-\infty, -\sqrt{\eta} \mathbf{P}_r] \cup [\sqrt{\eta} \mathbf{P}_r, \infty)$, if $i = 1$. The terms $\mathbb{E}[\mathbf{e}_c[t]]$ and $\text{Var}(\mathbf{e}_c[t])$ in the above can be in turn recursively computed as

$$\mathbb{E}[\mathbf{e}_c^{(i)}[t+1]] = \frac{\int_{a_i}^{b_i} \mathbb{E}[\mathbf{e}_d^{(i)}[t+1]] f_{\mathbf{y}}(y)}{\int_{a_i}^{b_i} f_{\mathbf{y}}(y)}; \mathbb{E}[\mathbf{e}_c[0]] = 0, i = 0, 1,$$

and $\text{Var}(\mathbf{e}_c^{(i)}[t+1]) = \mathbb{E}[(\mathbf{e}_c^{(i)}[t+1])^2] - (\mathbb{E}[\mathbf{e}_c^{(i)}[t+1]])^2$; $\text{Var}(\mathbf{e}_c[0]) = \mathbf{P}_e$. Finally the quantity $f_{\mathbf{i}_{[1:t]}}(\mathbf{b})$ can be computed recursively as $f_{\mathbf{i}_{[1:t+1]}}(\mathbf{b}) = \int_{a_i}^{b_i} f_{\mathbf{y}}(y), i = 0, 1$.

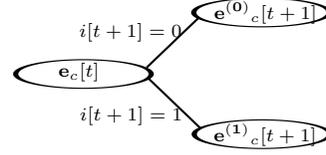


Fig. 14. Evolution of KF estimation error.

Appendix A - Part II

We now present the complete derivation of (9). First, we present a sketch of the derivation. From Fig. 14, note that $\mathbf{e}_c^{(0)}[t+1]$ can be equivalently expressed as

$$\mathbf{e}_c^{(0)}[t+1] = \mathbf{e}_c[t+1] | \{ \mathbf{i}_{[1:t]}, \mathbf{r}_c[t+1] \in [-\sqrt{\eta} \mathbf{P}_r, \sqrt{\eta} \mathbf{P}_r] \}.$$

We first derive an expression for $\mathbf{e}[t+1] | \{ \mathbf{i}_{[1:t]}, \mathbf{r}_c[t+1] = r \}$ where $r \in [-\sqrt{\eta} \mathbf{P}_r, \sqrt{\eta} \mathbf{P}_r]$. Then, we evaluate $\mathbb{E}[(\mathbf{e}_c^{(0)}[t+1])^2]$ by using $\mathbb{E}[X^2 | a \leq Y \leq b] = \frac{\int_a^b \mathbb{E}[X^2 | Y] f_Y(y)}{\int_a^b f_Y(y)}$, where $X = \mathbf{e}_c^{(0)}[t+1], Y = \mathbf{r}_c[t+1]$. The limits a and b correspond to $-\sqrt{\eta} \mathbf{P}_r$ and $\sqrt{\eta} \mathbf{P}_r$ respectively. We adopt a similar approach to evaluate $\mathbb{E}[(\mathbf{e}_c^{(1)}[t+1])^2]$. We present the detailed derivation in the following.

For brevity we only present analysis in the case when $\mathbf{r}_c[t+1] \in [-\sqrt{\eta} \mathbf{P}_r, \sqrt{\eta} \mathbf{P}_r]$. Recall the definitions of the terms $\mathbf{e}_c[t] = \mathbf{e}[t] | \mathbf{i}_{[1:t]}$, $\mathbf{r}_c[t] = \mathbf{r}[t] | \mathbf{i}_{[1:t-1]}$. They evolve in time as

$$\begin{aligned} \mathbf{e}_c[t+1] &= \mathbf{A}_K \mathbf{e}_c[t] + \mathbf{W}_K \mathbf{w}[t] \\ &\quad - \mathbf{K}(\mathbf{a}[t+1] - i[t+1] \boldsymbol{\delta}[t+1]) - \mathbf{K} \mathbf{v}[t+1], \quad t \geq 0, \end{aligned} \quad (13)$$

$$\mathbf{r}_c[t+1] = \mathbf{C} \mathbf{A} \mathbf{e}_c[t] + \mathbf{C} \mathbf{w}[t] + \mathbf{v}[t+1], \quad (14)$$

where the above are obtained by conditioning both sides of (6) and the residual dynamics $\mathbf{r}[t+1] = \mathbf{C} \mathbf{A} \mathbf{e}[t] + \mathbf{C} \mathbf{w}[t] + \mathbf{a}[t+1] + \mathbf{v}[t+1]$ on $\mathbf{i}_{[1:t]}$ and noting that $\mathbf{w}[t]$ and $\mathbf{v}[t]$ are independent of $\mathbf{i}_{[1:t]}$.

For convenience, let us define

$$\begin{aligned} \mathbf{e}_d^{(0)}[t+1] &\doteq \mathbf{e}[t+1] | \{ \mathbf{i}_{[1:t]}, \mathbf{r}_c[t+1] = r \} \\ &= \mathbf{A}_K \mathbf{e}_c[t] + \mathbf{W}_K \mathbf{w}[t] - \mathbf{K} \mathbf{a}[t+1] \\ &\quad - \mathbf{K} \mathbf{v}[t+1] | \{ \mathbf{r}_c[t+1] = r \}, \end{aligned} \quad (15)$$

where in the above we have used the definition of $\mathbf{e}_c[t]$ as well as the fact that $\mathbf{w}[t]$, and $\mathbf{v}[t+1]$ are independent of $\mathbf{i}_{[1:t]}, \mathbf{r}_c[t+1]$ and $i[t+1] = 0$. Let us denote $\mathbf{x} = \mathbf{A}_K \mathbf{e}_c[t] + \mathbf{W}_K \mathbf{w}[t] - \mathbf{K} \mathbf{a}[t+1] - \mathbf{K} \mathbf{v}[t+1]$ and $\mathbf{y} = \mathbf{r}_c[t+1]$. We note that \mathbf{x} and \mathbf{y} are Gaussian random variables and hence jointly Gaussian. Their mean and covariance matrix can be computed as

$$\begin{aligned} \mathbb{E}[\mathbf{x}] &= \mathbf{A}_K \mathbb{E}[\mathbf{e}_c[t]] - \mathbf{K} \mathbf{a}[t+1], \mathbb{E}[\mathbf{y}] \\ &= \mathbf{C} \mathbf{A} \mathbb{E}[\mathbf{e}_c[t]] + \mathbf{a}[t+1] \quad \text{and} \quad \text{Cov}(\mathbf{X}) = \begin{bmatrix} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xy}} \\ \Sigma_{\mathbf{yx}} & \Sigma_{\mathbf{yy}} \end{bmatrix}, \end{aligned}$$

where terms $\Sigma_{\mathbf{xx}}, \Sigma_{\mathbf{xy}}$ and $\Sigma_{\mathbf{yy}}$ are defined in Appendix A, Part I (for $i = 0$).

We now present in recursive method to compute $\mathbb{E}[\mathbf{e}_c[t]]$ in the above expression. First note based on our definitions, $\mathbf{e}_d^{(0)}[t+1]$ is the conditional random variable $\mathbf{x}|\mathbf{y}$. Thus, the mean and variance of $\mathbf{e}_d^{(0)}[t+1]$ can be derived using multivariate normal distribution as

$$\mathbb{E}[\mathbf{e}_d^{(0)}[t+1]] = \mathbb{E}[\mathbf{x}] + \Sigma_{\mathbf{x}\mathbf{y}}\Sigma_{\mathbf{y}\mathbf{y}}^{-1}(\mathbf{y} - \mathbb{E}[\mathbf{y}]), \quad (16)$$

$$\text{Var}(\mathbf{e}_d^{(0)}[t+1]) = \Sigma_{\mathbf{x}\mathbf{x}} - \Sigma_{\mathbf{x}\mathbf{y}}\Sigma_{\mathbf{y}\mathbf{y}}^{-1}\Sigma_{\mathbf{y}\mathbf{x}}. \quad (17)$$

From (16), we can compute the $\mathbb{E}[\mathbf{e}_c^{(0)}[t+1]]$ as

$$\mathbb{E}[\mathbf{e}_c^{(0)}[t+1]] = \frac{\int_{-\sqrt{\eta}\mathbf{P}_r}^{\sqrt{\eta}\mathbf{P}_r} \mathbb{E}[\mathbf{e}_d^{(0)}[t+1]]f_{\mathbf{y}}(\mathbf{y})}{\int_{-\sqrt{\eta}\mathbf{P}_r}^{\sqrt{\eta}\mathbf{P}_r} f_{\mathbf{y}}(\mathbf{y})}, \quad (18)$$

where we have used the fact that $\mathbb{E}[X|a \leq Y \leq b] = \frac{\int_a^b \mathbb{E}[X|Y]f_Y(y)}{\int_a^b f_Y(y)}$. A similar method can be used to compute

$\mathbb{E}[\mathbf{e}_c^{(1)}[t+1]]$. Then $\mathbb{E}[\mathbf{e}_c[t+1]]$ can be computed as $\mathbb{E}[\mathbf{e}_c[t+1]] = \mathbb{P}(i[t+1] = 0)\mathbb{E}[\mathbf{e}_c^{(0)}[t+1]] + \mathbb{P}(i[t+1] = 1)\mathbb{E}[\mathbf{e}_c^{(1)}[t+1]]$.

The terms (16) and (17) can also be used to compute $\mathbb{E}[(\mathbf{e}_c^{(0)}[t+1])^2]$ as

$$\mathbb{E}[(\mathbf{e}_c^{(0)}[t+1])^2] = \frac{\int_{-\sqrt{\eta}\mathbf{P}_r}^{\sqrt{\eta}\mathbf{P}_r} \mathbb{E}[(\mathbf{e}_d^{(0)}[t+1])^2]f_{\mathbf{y}}(\mathbf{y})}{\int_{-\sqrt{\eta}\mathbf{P}_r}^{\sqrt{\eta}\mathbf{P}_r} f_{\mathbf{y}}(\mathbf{y})}, \quad (19)$$

where in we used the fact that $\mathbb{E}[X^2|a \leq Y \leq b] = \frac{\int_a^b \mathbb{E}[X^2|Y]f_Y(y)}{\int_a^b f_Y(y)}$. Note that integrand of (19) is given by

$$\mathbb{E}[(\mathbf{e}_d^{(0)}[t+1])^2] = \left(\mathbb{E}[\mathbf{e}_d^{(0)}[t+1]]\right)^2 + \text{Var}(\mathbf{e}_d^{(0)}[t+1]),$$

where the RHS terms are in (16) and (17) respectively. We remark that the case $\mathbf{r}_c[t+1] \notin [-\sqrt{\eta}\mathbf{P}_r, \sqrt{\eta}\mathbf{P}_r]$ can be similarly analyzed to derive $\mathbb{E}[(\mathbf{e}_c^{(1)}[t+1])^2]$. We skip the derivation here and The final result is stated in Appendix A, Part I.

Finally, the quantity $f_{\mathbf{i}_{[1:t]}}(\mathbf{b})$ can be computed by noting that $\mathbb{P}(i[t+1] = 0|\mathbf{i}_{[1:t]}) = \mathbb{P}(\mathbf{r}_c[t+1] \in [-\sqrt{\eta}\mathbf{P}_r, \sqrt{\eta}\mathbf{P}_r])$ and $\mathbb{P}(i[t+1] = 1|\mathbf{i}_{[1:t]}) = \mathbb{P}(\mathbf{r}_c[t+1] \notin [-\sqrt{\eta}\mathbf{P}_r, \sqrt{\eta}\mathbf{P}_r])$. The quantities $\mathbb{P}(\mathbf{r}_c[t+1] \in [-\sqrt{\eta}\mathbf{P}_r, \sqrt{\eta}\mathbf{P}_r])$ and $\mathbb{P}(\mathbf{r}_c[t+1] \notin [-\sqrt{\eta}\mathbf{P}_r, \sqrt{\eta}\mathbf{P}_r])$ can be computed by noting that $\mathbf{r}_c[t+1]$ is a Gaussian distributed random variable from with mean $\mathbb{E}[\mathbf{y}]$ and variance $\Sigma_{\mathbf{y}\mathbf{y}}$ (from (14)).



Subhash Lakshminarayana (S'07-M'12) is an assistant professor at the School of Engineering, University of Warwick, UK. Previously, he worked as a researcher at the Advanced Digital Sciences Center (ADSC) in Singapore between 2015-2018, a joint post-doctoral researcher at Princeton University and the Singapore University of Technology and Design (SUTD) between 2013-2015. He received his Ph.D. from the Alcatel Lucent Chair on Flexible Radio and the Department of Telecommunications at SU-PELEC, France in 2012, M.S. degree in Electrical and Computer Engineering from The Ohio State University in 2009 and B.S. from Bangalore University, India.

His research interests include cyber-physical system security (power grids and urban transportation) and wireless communications. His works have been selected among the Best conference papers on integration of renewable & intermittent resources at the IEEE PESGM - 2015 conference, and the "Best 50 papers" of IEEE Globecom 2014 conference.



Jabir Shabbir Karachiwala obtained Masters degree in Computing from School of Computing in National University Of Singapore in 2015. Between 2016-2018, he worked as research engineer and software engineer at Advanced Digital Science Centre and Singapore MIT Alliance Of Research And Technology in Singapore on topics pertaining to transportation and cybersecurity. Currently he is working as Member Of technical Staff at Oracle India Pvt Ltd.



Teo Zhan Teng was a research engineer at the Advanced Digital Sciences Center, Singapore from 2015-2016. Currently, he is working with GovTech, Singapore.



Rui Tan (M'08-SM'18) is an Assistant Professor at School of Computer Science and Engineering, Nanyang Technological University, Singapore. Previously, he was a Research Scientist (2012-2015) and a Senior Research Scientist (2015) at Advanced Digital Sciences Center, a Singapore-based research center of University of Illinois at Urbana-Champaign (UIUC), a Principle Research Affiliate (2012-2015) at Coordinated Science Lab of UIUC, and a postdoctoral Research Associate (2010-2012) at Michigan State University. He received the Ph.D.

(2010) degree in computer science from City University of Hong Kong, the B.S. (2004) and M.S. (2007) degrees from Shanghai Jiao Tong University. His research interests include cyberphysical systems, sensor networks, and pervasive computing systems. He received the Best Paper Awards from IPSN'17, CPSR-SG'17, Best Paper Runner-Ups from IEEE PerCom'13 and IPSN'14.



David K.Y. Yau received the B.Sc. from the Chinese University of Hong Kong, and M.S. and Ph.D. from the University of Texas at Austin, all in computer science. He has been Professor at Singapore University of Technology and Design since 2013. Since 2010, he has been Distinguished Scientist at the Advanced Digital Sciences Centre, Singapore. He was Associate Professor of Computer Science at Purdue University (West Lafayette). He received an NSF CAREER award. He won Best Paper award in 2017 ACM/IEEE IPSN and 2010 IEEE MFI. His

papers in 2008 IEEE MASS, 2013 IEEE PerCom, 2013 IEEE CPSNA, and 2013 ACM BuildSys were Best Paper finalists. His research interests include cyber-physical system and network security/privacy, wireless sensor networks, smart grid IT, and quality of service. He serves as Associate Editor of IEEE Trans. Network Science and Engineering and ACM Trans. Sensor Networks. He was Associate Editor of IEEE Trans. Smart Grid, Special Section on Smart Grid CyberPhysical Security (2017), IEEE/ACM Trans. Networking (2004-09), and Springer Networking Science (2012-2013); Vice General Chair (2006), TPC co-Chair (2007), and TPC Area Chair (2011) of IEEE ICNP; TPC co-Chair (2006) and Steering Committee member (2007-09) of IEEE IWQoS; TPC Track co-Chair of 2012 IEEE ICDCS; and Organizing Committee member of 2014 IEEE SECON. He is a senior member of IEEE.