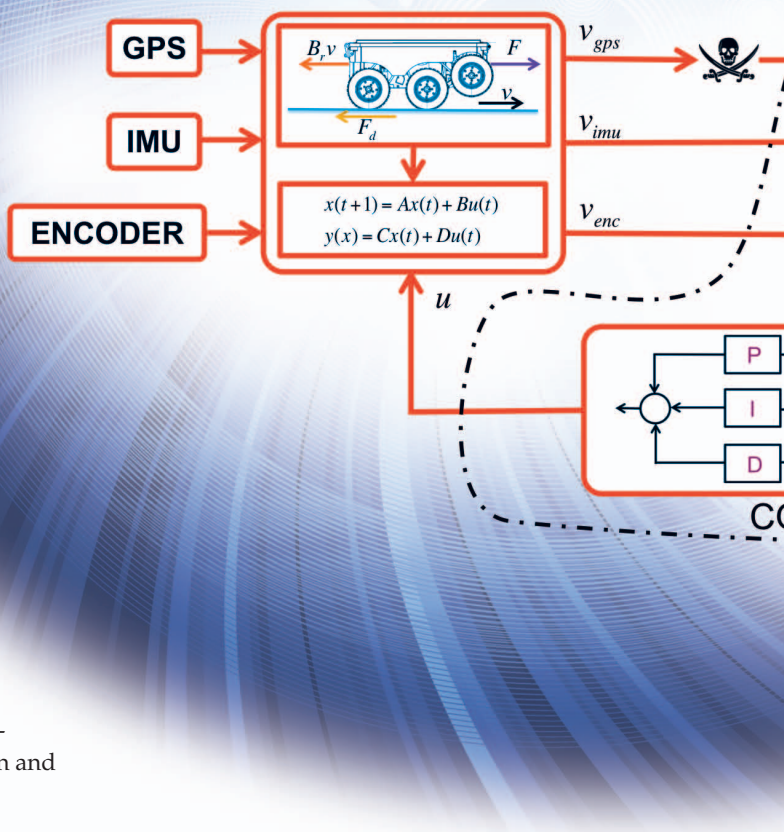
Design and Implementation of Attack-Resilient Cyberphysical Systems

WITH A FOCUS ON ATTACK-RESILIENT STATE ESTIMATORS

MIROSLAV PAJIC, JAMES WEIMER, NICOLA BEZZO, OLEG SOKOLSKY, GEORGE J. PAPPAS, and INSUP LEE



Recent years have witnessed a significant increase in the number of security-related incidents in control systems. These include high-profile attacks in a wide range of application domains, from attacks on critical infrastructure, as in the case of the Maroochy Water breach [1], and industrial systems (such as the StuxNet virus attack on an industrial supervisory control and data acquisition system [2], [3] and the German Steel Mill cyberattack [4], [5]), to attacks on modern vehicles [6]–[8]. Even high-assurance military systems were shown to be vulnerable to attacks, as illustrated in the highly publicized downing of the RQ-170 Sentinel U.S. drone [9]–[11]. These incidents have greatly raised awareness of the need for security in cyberphysical systems (CPSs), which feature tight coupling of computation and

Digital Object Identifier 10.1109/MCS.2016.2643239
Date of publication: 17 March 2017

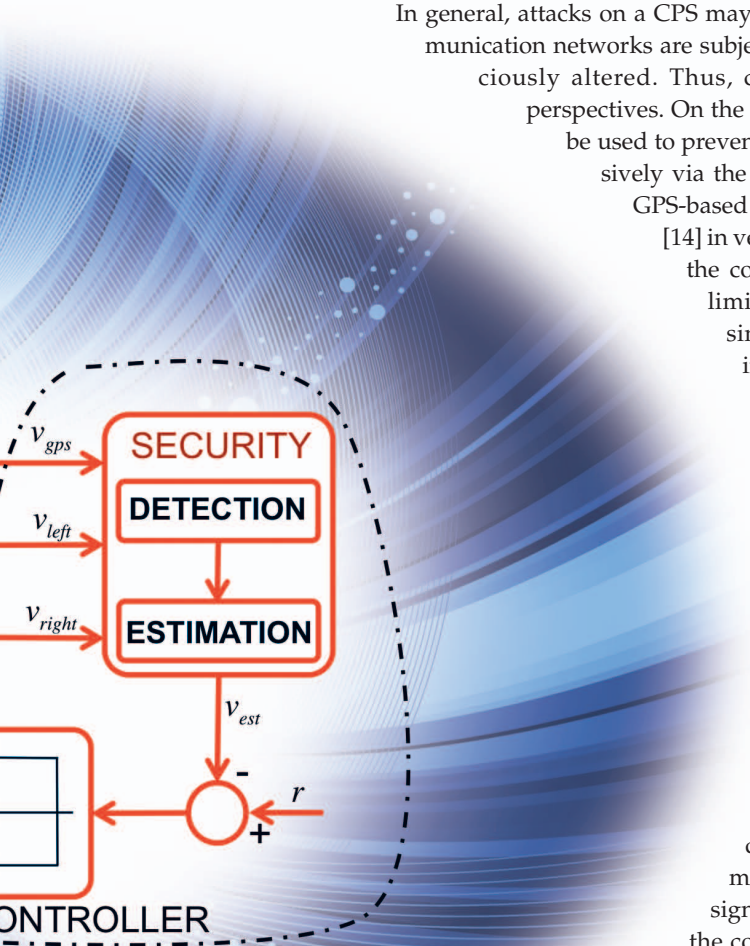
BACKGROUND ART—ALL-FREE-DOWNLOAD.COM

1066-033X/17©2017IEEE

communication substrates with sensing and actuation components. However, the complexity and heterogeneity of this next generation of safety-critical, networked, and embedded control systems have challenged the existing design methods in which security is usually consider as an afterthought.

This is well illustrated in modern vehicles, which present a complex interaction of a large number of embedded electronic control units, communicating over an internal network or multiple networks. On the one hand, there is a current shift in vehicle architectures, from isolated control systems to more open automotive architectures with services such as remote diagnostics and code updates and vehicle-to-vehicle communication. On the other hand, this increasing set of functionalities, network interoperability, and system design complexity may introduce security vulnerabilities that are easily exploitable. Security guarantees for these systems are usually based on perimeter security where internal networks are resource constrained, mostly depending on the security of the gateway and external communication channels. Thus, any successful attacks on the gateway or external communication, or physical attacks on components connected to an internal network, could completely compromise the system; using simple methods an attacker can disrupt the operation of a car, even taking complete control over it, as shown in [6]–[8].

In general, attacks on a CPS may affect all of its components; computational nodes and communication networks are subject to intrusions, and the physical environment may be maliciously altered. Thus, control-specific CPS security challenges arise from two perspectives. On the one hand, conventional information-security approaches can be used to prevent intrusions, but attackers can still affect the system noninvasively via the physical environment. For instance, noninvasive attacks on GPS-based navigation systems [11]–[13] and antilock braking systems [14] in vehicles illustrate how an adversarial signal can be injected into the control loop using the sensor measurements. This highlights limitations of the standard cyber-based security mechanisms since, even if employed communication protocols over the internal networks ensure data integrity, they alone do not guarantee resilience of control systems to attacks on the system's physical components. On the other hand, getting access to an internal network would allow the attacker to compromise sensors to controller to actuators communication; from the control perspective, these attacks can also be modeled as additional adversary signals introduced via the sensors and actuators [15]. Although these types of attacks could be addressed with the use of cryptographic tools that guarantee data integrity, resource constraints inherent in many CPS domains may prevent heavy-duty security approaches from being deployed.



Therefore, it is necessary to address the security challenge related to the attacks against the control system, where the attacker can 1) take over a sensor and supply wrong or delayed sensor readings or 2) disrupt actuation. These attacks manifest themselves to the controller as malicious interference signals, and the defenses against them have to be introduced in the control-design phase. Specifically, resilience against these attacks is built into the control algorithm under the assumption that the controller itself executes according to its specification. This approach has attracted a lot of attention, with several efforts focused on the use of control-level techniques that exploit a model of the "normal" system behavior, for attack detection and identification in CPS (see, for instance, [15]–[22]). Methods for attack detection based on the use of standard residual-probability-based detectors were presented in [21]–[24], while the problem of state estimation in the presence of sensors attacks was addressed in [17], [18], [25], and [26].

In contrast, attacks on the execution platform prevent the correct operation of the control system, as in the cases where the attacker can disrupt the execution of control tasks. Defense against such

The problem of state estimation in the presence of sensor and actuator attacks has attracted significant attention in recent years.

attacks cannot rely on the control algorithm, which may not be running correctly. Instead, it requires security and performance guarantees that the platform components provide to the control system, which have to be incorporated into the design of control-based security techniques. For example, the attacker may try to affect control performance by dramatically slowing down the controller task; one way to achieve this is by introducing a higher-priority, computationally intensive task into the operating system. The key to addressing these types of attacks is to explicitly specify the assumptions made about the platform during the control design. Real-time issues, such as sampling and actuation jitter, and synchronization errors between system components, directly affect quality of control and the level of guarantees provided by control-based security mechanisms. For instance, execution timing directly affects the controlled plant's model that should be used for control-level security techniques; control engineers may determine that the controller guarantees the required resiliency levels (for instance, attack detection) and the desired control performance, as long as the worst-case execution time of the control task is, for example, 20 ms and output jitter is no more than 2 ms.

For attack-resilient control in CPSs, it is necessary to be able to capture platform effects on the control-level security guarantees by providing robust security-aware control methods that can deal with noise and modeling errors. This will enable the extraction of system-level requirements imposed by control algorithms on the underlaying operating system (OS) and network and facilitate reasoning about attack resilience across different implementation layers.

This article describes efforts on the development of an attack-resilient CPS. Specifically, a case study is considered, the design of a resilient cruise controller for an autonomous ground vehicle, focusing on one component of the system, namely an attack-resilient state estimator and the performance guarantees in the presence of attacks. The article starts by addressing the problem of attack-resilient state estimation, before providing robustness guarantees for the implemented attack-resilient state estimator (building on [25]). It is shown that the maximal performance loss imposed by a smart attacker, exploiting the difference between the model used for state estimation and the *real* physical dynamics of the system, is bounded and linear with the size of the noise and modeling errors. Furthermore, this article describes how implementation issues such as jitter, latency, and synchronization errors can be mapped to parameters of the state-estimation procedure, which effectively enables mapping control performance requirements to real time (that is, timing related) specifications imposed on the underlying platform. Finally, how to construct an assurance case for the system that covers both a mathematical model of the state estimator and the physical environment is presented as well as a software implementation of the controller. While the models considered in the case study are specific to the control system and its intended deployment platform, the modeling, robustness analysis, and assumptions encountered on each level in this case study are typical of many other CPS control problems.

ATTACK-RESILIENT STATE ESTIMATION WITH NOISE AND MODELING ERRORS

The problem of state estimation in the presence of sensor and actuator attacks has attracted significant attention in recent years. One motivation is that the same controllers can be used when there is no attack, provided that the controller can obtain a reasonable estimate of the state of the physical process even if some of the sensor measurements and actuator commands have been compromised. For deterministic (that is, noiseless) linear time-invariant (LTI) systems, the correct state estimate in the presence of sensor attacks can be obtained as the solution of l_0 optimization problems [17], [18]. In addition, there are estimation techniques for linear [26] and differentially flat systems [27] based on the use of satisfiability modulo theories solvers.

However, the initially proposed techniques for state estimation in the presence of attacks focus on noiseless systems for which the exact model of the system's dynamics is known. Hence, these techniques have limited applicability to real systems since it is unclear what level of resiliency can be guaranteed with more realistic sensing, actuation, and execution models. Therefore, the focus of this section is on attack-resilient state estimation for dynamical systems with bounded noise and modeling errors and the derivation of a worst-case bound for performance degradation in the presence of attacks. First, the system model and how some implementation effects can be mapped into the model's parameters are presented. Next, the estimator and the procedure to bound the worst-case estimation error in the presence of attacks is introduced.

Notation and Terminology

In this article, $|\mathcal{S}|$ denotes the cardinality (size) of the set $\mathcal{S}$. For two sets $\mathcal{S}$ and $\mathcal{R}$, $\mathcal{S}/\mathcal{R}$ denotes the set of elements in $\mathcal{S}$ that are not in $\mathcal{R}$. For a set $\mathcal{K} \subset \mathcal{S}$, $\mathcal{K}^{\complement}$ specifies the complement set of $\mathcal{K}$ with respect to $\mathcal{S}$, that is, $\mathcal{K}^{\complement} = \mathcal{S}/\mathcal{K}$. Also, $\mathbb{R}$ is used to denote the set of reals and $1'_N$ to denote

the row vector of size N containing all ones. Finally, for any sequence of α_i, $i \geq 0$, since the sum $\sum_0^{-1} \alpha_i$ contains no elements, to simplify the notation it is assumed that it is equal to zero, that is, $\sum_0^{-1} \alpha_i = 0$.

The ith element of a vector $\mathbf{x}_k$ is denoted by $\mathbf{x}_{k,i}$. For vector $\mathbf{x}$ and matrix $\mathbf{A}$, $|\mathbf{x}|$ and $|\mathbf{A}|$ denote the vector and matrix whose elements are absolute values of the initial vector and matrix, respectively. Also, for matrices $\mathbf{P}$ and $\mathbf{Q}$, $\mathbf{P} \preceq \mathbf{Q}$ is used to specify that the matrix $\mathbf{P}$ is *element wise* smaller than the matrix $\mathbf{Q}$. For a vector $\mathbf{e} \in \mathbb{R}^p$, *the support* of the vector is set

$$\text{supp}(\mathbf{e}) = \{i \,|\, \mathbf{e}_i \neq 0\} \subseteq \{1,2,...,p\},$$

while the l_0 norm of vector $\mathbf{e}$ is the size of $\text{supp}(\mathbf{e})$, that is, $\|\mathbf{e}\|_{l_0} = |\text{supp}(e)|$. Note that, although l_0 is not formally a norm, in this article we will abuse the terminology and refer to it as a norm to maintain consistency with the terminology used in previous work on this topic (for example, [18]). Also, for a matrix $\mathbf{E} \in \mathbb{R}^{p \times N}$, $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N$ is used to denote its columns and $\mathbf{E}'_1, \mathbf{E}'_2, ..., \mathbf{E}'_p$ to denote its rows. The *row support* of matrix $\mathbf{E}$ is defined as the set

$$\text{rowsupp}(\mathbf{E}) = \{i \,|\, \mathbf{E}'_i \neq 0\} \subseteq \{1,2,...,p\}.$$

As for vectors, the l_0 norm for a matrix $\mathbf{E}$ is defined as $\|\mathbf{E}\|_{l_0} = |\text{row supp}(\mathbf{E})|$.

SYSTEM MODEL

This article considers an LTI system

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{v}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{w}_k + \mathbf{e}_k, \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^m$ denote the plant's state and input vectors, respectively, while $\mathbf{y} \in \mathbb{R}^p$ is the plant's output vector obtained from measurements of p sensors from the set $\mathcal{S} = \{1,2,...,p\}$. Accordingly, the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ have suitable dimensions. Furthermore, $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{w} \in \mathbb{R}^p$ denote the process and measurement noise vectors, while $\mathbf{e} \in \mathbb{R}^p$ denotes the attack vector. The set $\mathcal{K} \subseteq \{1,2,...,p\}$, containing sensors under attack, is used to model attacks on plant sensors. This means that $\mathbf{e}_{k,i} = 0$ for all $i \in \mathcal{K}^C$ and $k \geq 0$, where $\mathcal{K}^C = \mathcal{S}/\mathcal{K}$, and therefore $\text{supp}(\mathbf{e}_k) \subseteq \mathcal{K}$ for all $k \geq 0$. This work assumes that the noise vectors are constrained in certain ways. Furthermore, $\mathbf{v}$ and $\mathbf{w}$ are used to capture different types of modeling errors that may be caused by some implementation (such as real-time) issues.

Note that the setup presented in this article can be easily extended to include attacks on the system's actuators. In this case, an additional vector e_k^a is added to the plant input at each step $k \geq 0$. As shown in [18], the same technique used for resilient-state estimation in the presence of attacks on sensors can be used to obtain the plant's state when subsets of the plant's sensors and actuators are both compromised. Consequently, the analysis and results presented here can be easily extended to the case when a subset of the actuators is also under attack. It is important to highlight that, in cases where a small enough subsets of plant actuators and sensors are compromised (that is, enabling the resilient state estimation), even with accurate estimates of the plant's state, stability cannot be guaranteed due to attacks on actuators, and the attacker could effectively gain complete control over the plant. This is consistent with the results from [16].

Attack-Resilient State Estimation for Noiseless Dynamical Systems

For linear systems without noise (that is, systems in the form (1) where $\mathbf{w}_k = 0$ and $\mathbf{v}_k = 0$, for all $k \geq 0$), an l_0-norm-based method to extract state estimates in the presence of attacks is introduced in [18]. To obtain the plant's state at any time-step t (that is, $\mathbf{x}_t$), the proposed procedure uses the previous N sensor measurement vectors $(\mathbf{y}_{t-N+1}, ..., \mathbf{y}_t)$ and actuator inputs $(\mathbf{u}_{t-N+1}, ..., \mathbf{u}_{t-1})$ to evaluate the state $\mathbf{x}_{t-N+1}$. The state $\mathbf{x}_t$ is then computed using the history of actuator inputs $(\mathbf{u}_{t-N+1}, ..., \mathbf{u}_{t-1})$ by applying the system evolution from (1) for $N-1$ steps. Specifically, the state $\mathbf{x}_{t-N+1}$ is computed as the minimization argument of the following optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Y}_{t,N} - \mathbf{\Phi}_N(\mathbf{x})\|_{l_0}. \tag{2}$$

Here, $\mathbf{Y}_{t,N} = [\tilde{\mathbf{y}}_{t-N+1} \,|\, \tilde{\mathbf{y}}_{t-N+2} \,|\, ... \,|\, \tilde{\mathbf{y}}_t] \in \mathbb{R}^{p \times N}$ aggregates the last N sensor measurements while taking into account the inputs applied during that interval

$$\begin{aligned} \tilde{\mathbf{y}}_k &= \mathbf{y}_k, && k = t-N+1, \\ \tilde{\mathbf{y}}_k &= \mathbf{y}_k - \sum_{i=0}^{k-t+N-2} \mathbf{C}\mathbf{A}^i\mathbf{B}\mathbf{u}_{k-1-i}, && k = t-N+2, ..., N. \end{aligned}$$

Furthermore, $\mathbf{\Phi}_N: \mathbb{R}^n \to \mathbb{R}^{p \times N}$ is a linear mapping defined as $\mathbf{\Phi}_N(\mathbf{x}) = [\mathbf{C}\mathbf{x} \,|\, \mathbf{C}\mathbf{A}\mathbf{x} \,|\, ... \,|\, \mathbf{C}\mathbf{A}^{N-1}\mathbf{x}]$ that captures the system's evolution over N steps caused by the initial state $\mathbf{x}$.

The rationale behind problem (2) is that the matrix $\mathbf{E}_{t,N} = \mathbf{Y}_{t,N} - \mathbf{\Phi}_N(\mathbf{x}_{t-N+1})$ presents the history of the last N attack vectors $\mathbf{e}_{t-N+1}, ..., \mathbf{e}_t$, that is,

$$\mathbf{E}_{t,N} = [\mathbf{e}_{t-N+1} \,|\, \mathbf{e}_{t-N+2} \,|\, ... \,|\, \mathbf{e}_t] \in \mathbb{R}^{p \times N}. \tag{3}$$

The critical observation here is that for a noiseless LTI system there is a pattern of zeros (that is, zero rows) in the matrix $\mathbf{E}_{t,N}$ that corresponds to the nonattacked sensors and which remains constant over time. If $\mathcal{K}$ is the set of compromised sensors, then, for all N, t such that $N \geq 0$, $t \geq N-1$, $\text{rowsupp}(\mathbf{E}_{t,N}) \subseteq \mathcal{K}$.

As shown in [18], for noiseless systems, the state estimator from (2) is optimal in the sense that if another estimator can recover $\mathbf{x}_{t-N+1}$, then the one defined in (2) can as well. In addition, the estimator from (2) can extract the system's

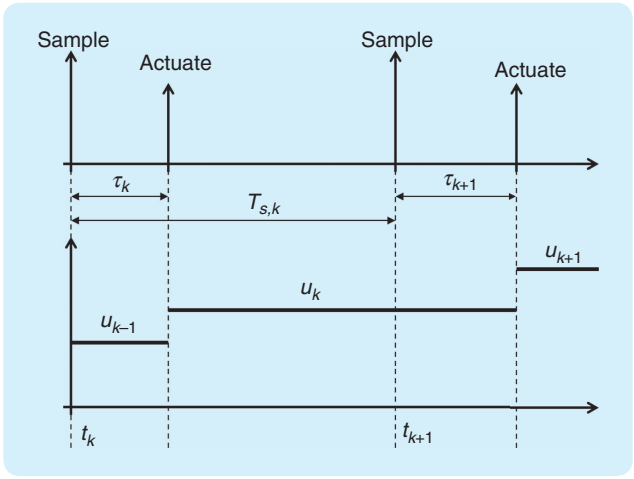


FIGURE 1 Scheduling sampling and actuation.

state after N steps when up to q sensors are under attack if and only if, for all $\mathbf{x} \in \mathbb{R}\backslash\{0\}$,

$$|\text{supp}(\mathbf{Cx}) \cup \text{supp}(\mathbf{CAx}) \cup \cdots \cup \text{supp}(\mathbf{CA}^{N-1}\mathbf{x})| > 2q.$$

In this article, $q_{\max}$ denotes the maximal number of compromised sensors for which the system's state can be recovered after N steps despite attacks on sensors. However, note that the size of the measurement history N is considered to be an input parameter to the resilient-state estimator. In the general case, the notation $q_{\max,N}$ should be used. Hence, if the number of compromised sensors q satisfies $q \leq q_{\max}$, for noiseless systems the minimal l_0 norm of (2) is equal to q. In addition, note that for these systems $q_{\max}$ does not decrease with N, and, due to the Cayley–Hamilton theorem [28], it cannot be further increased when more than n previous measurements are used, that is, $q_{\max}$ obtains the maximal value for $N = n$. Finally, in addition to the measurement window size N, $q_{\max}$ only depends on the system's dynamics (that is, matrices $\mathbf{A}$ and $\mathbf{C}$), as was characterized in [18] and [29]. To formally capture this dependency, consider the following notation. For any set $\mathcal{K} = \{k_1, \ldots, k_{|\mathcal{K}|}\} \subseteq S$, where $k_1 < k_2 < \cdots < k_{|\mathcal{K}|}$, the matrices $\mathbf{O}_{\mathcal{K}}$ and $P_{\mathcal{K}}$ are

$$\mathbf{O}_{\mathcal{K}} = \begin{bmatrix} P_{\mathcal{K}}\mathbf{C} \\ P_{\mathcal{K}}\mathbf{CA} \\ \vdots \\ P_{\mathcal{K}}\mathbf{CA}^{N-1} \end{bmatrix}, \quad P_{\mathcal{K}} = \begin{bmatrix} \mathbf{i}'_{k_1} \\ \vdots \\ \mathbf{i}'_{k_{|\mathcal{K}|}} \end{bmatrix}. \tag{4}$$

Here, $P_{\mathcal{K}}$ denotes the projection from the set $\mathcal{S}$ to the set $\mathcal{K}$ by keeping only rows of $\mathbf{C}$ with indices that correspond to sensors from $\mathcal{K}$, because $\mathbf{i}'_j$ denotes the row vector (of appropriate size) with a one in its jth position.

Definition 1 [29]

An LTI system (1) is said to be s-sparse observable if, for every set $\mathcal{K} \subset \mathcal{S}$ of size s (that is, $|\mathcal{K}| = s$), the pair $(\mathbf{A}, P_{\mathcal{K}^c}\mathbf{C})$ is observable.

The following lemma holds from results in [18] and [29].

Lemma 1

$q_{\max}$ is equal to the maximal s for which the system is $2s$-sparse observable.

Sources of Modeling Errors

Beside process and measurement noise, vectors $\mathbf{v}_k$ and $\mathbf{w}_k$ in (1) can be used, in some cases, to capture deviations in the plant model from the real dynamics of the controlled physical system. One source of modeling errors is the impossibility of obtaining accurate parameter values during initial modeling experiments. In the general case, these types of errors are dominant in the overall model error. However, in some cases, significant modeling errors are introduced by nonidealities of control system implementation and limitations of the used computation and communication platforms. For instance, modeling errors can be caused by sampling and computation/actuation jitter and synchronization errors between system components in scenarios where continuous-time plants are being controlled. Errors of this type are emphasized in control systems in which underlying computation and communication platforms provide very loose execution guarantees.

The described attack-resilient state estimator (2) is based on the discrete-time model (1) of the system. Consequently, to be able to deal with continuous-time plants, it is convenient to discretize the controlled plant, while taking into account real-time issues introduced by communication and computation schedules. To illustrate this, consider a standard continuous-time plant model

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}_c\mathbf{x}(t), \end{aligned} \tag{5}$$

with state $\mathbf{x}(t) \in \mathbb{R}^n$, output $\mathbf{y}(t) \in \mathbb{R}^p$, and input vector $\mathbf{u}(t) \in \mathbb{R}^m$, where matrices $\mathbf{A}_c, \mathbf{B}_c$, and $\mathbf{C}_c$ are of the appropriate dimensions.

First, consider setups where all the plant's outputs are sampled (that is, measured) at times t_k, $k \geq 0$ and where all actuators apply newly calculated inputs at times $t_k + \tau_k$, $k \geq 0$, as shown in Figure 1. Here, the kth sampling period of the plant is denoted by $T_{s,k} = t_{k+1} - t_k$, and the input signal will have the form shown in the lower diagram of Figure 1. Using the approach from [30] and [31], the system can be described as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}_c\mathbf{x}(t), \quad t \in [t_k + \tau_k, t_{k+1} + \tau_{k+1}), \\ \mathbf{u}(t^+) &= \mathbf{u}_k, \quad t \in \{t_k + \tau_k, k = 0, 1, 2, \ldots\}, \end{aligned} \tag{6}$$

where $\mathbf{u}(t^+)$ is a piecewise-continuous function that only changes values at time instances $t_k + \tau_k$, $k \geq 0$. Thus, the discretized system model can be represented as [28]

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{B}_k^-\mathbf{u}_{k-1}, \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k, \end{aligned} \tag{7}$$

where $\mathbf{x}_k = \mathbf{x}(t_k)$, $k \geq 0$, and

$$\mathbf{A}_k = e^{\mathbf{A}_c T_{s,k}},$$
$$\mathbf{B}_k = \int_0^{T_{s,k}-\tau_k} e^{\mathbf{A}_c\theta}\mathbf{B}_c d\theta,\ \mathbf{B}_k^- = \int_{T_{s,k}-\tau_k}^{T_{s,k}} e^{\mathbf{A}_c\theta}\mathbf{B}_c d\theta. \tag{8}$$

Note that the matrices $\mathbf{A}_k, \mathbf{B}_k$, and $\mathbf{B}_k^-$ are time varying (with k) and depend on the continuous-time plant dynamics, intersampling time $T_{s,k}$, and latency τ_k. On the other hand, when control (and state estimation) is performed using resource-constrained CPUs, the designers usually use the "ideal" discrete-time model of the system of the form (1) where, for all $k \geq 0$, $T_{s,k} = T_s$, and $\tau_k = 0$,

$$\mathbf{A} = e^{\mathbf{A}_c T_s} \text{ and } \mathbf{B} = \int_0^{T_s} e^{\mathbf{A}_c\theta}\mathbf{B}_c d\theta. \tag{9}$$

Hence, by comparing the discrete-time models (1) and (7), in this case, sampling and actuation jitter and actuation latency (caused by computation and/or communication) introduce the error component $\mathbf{v}_k^{jit}$ $(k \geq 0)$ defined as

$$\mathbf{v}_k^{jit} = \underbrace{(e^{\mathbf{A}_c T_{s,k}} - e^{\mathbf{A}_c T_s})}_{\triangle\mathbf{A}}\mathbf{x}_k + \underbrace{\int_{T_s}^{T_{s,k}-\tau_k} e^{\mathbf{A}_c\theta}\mathbf{B}_c d\theta}_{\triangle\mathbf{B}}\mathbf{u}_k + \mathbf{B}_k^-\mathbf{u}_{k-1}. \tag{10}$$

Finally, from (10) it follows that a bound on the size of the error $\mathbf{v}_k^{jit}$ can obtained from the conservative bounds on the sampling jitter (that is, $T_{s,k} - T_s$) and latency (τ_k), for a predefined range of acceptable system states and actuator inputs. For example, boundedness of the system state can be ensured in the case where the actual closed-loop system is stable.

Effects of Synchronization Errors

To simplify the presentation, only systems where the sensors do not have a common clock source are considered, that is, where there possibly exist synchronization errors between sensors; the same approach can be extended to scenarios with synchronization errors between plant actuators. In this case, although scheduled to sample at the same time instance t_k, each sensor j will actually perform measurement at time $t_{k,j}$. Therefore, for every $j = 1,...,p$, $\mathbf{y}_{k,j} = \mathbf{C}_j'\mathbf{x}(t_{k,j})$ instead of $\mathbf{C}_j'\mathbf{x}(t_k)$, where $\mathbf{C}_j'$ denotes the jth row of C, meaning that the synchronization error introduces a measurement error defined as

$$\begin{aligned}\mathbf{v}_{k,j}^{\text{syn}} &= \mathbf{C}_j'(\mathbf{x}(t_k) - \mathbf{x}(t_{k,j})) \\ &= \mathbf{C}_j'\left(e^{\mathbf{A}_c\Delta t_{k,j}}\mathbf{x}(t_k) + \int_0^{\Delta t_{k,j}} e^{\mathbf{A}_c\theta}\mathbf{B}_c d\theta\mathbf{u}_{k-1}\right).\end{aligned} \tag{11}$$

Here, $\Delta t_{k,j} = t_k - t_{k,j}$ captures the synchronization error for each sensor j. Hence, if the plant state can be bounded (for example, due to closed-loop system stability), for a predefined actuation range it is possible to provide a bound on the size of the measurement error vector $\mathbf{v}_k^{\text{syn}} \in \mathbb{R}^p$ describing modeling errors caused by synchronization errors between sensors.

l_0-BASED METHOD FOR RESILIENT STATE ESTIMATION IN THE PRESENCE OF NOISE

In the rest of this section, unless otherwise specified, the term *noise* will be used to both include process and measurement noise and capture modeling errors, that is, the discrepancy between the model used to design the state estimator and the real dynamics of the plant. The presence of noise limits the use of the attack-resilient state estimator from (2). For example, in this case, the l_0 norm of a solution of the problem in (2) may be larger than $q_{\max}$, indicating that more than the allowed number of sensors has been compromised, which violates requirements for correct operation of the state estimator. Therefore, it is necessary to provide a method for attack-resilient state estimators in the presence of noise, with a provable bound on the worst-case performance degradation of the introduced resilient-state estimator due to the bounded size noise.

As illustrated in the previous section, the effects of the input vectors $\mathbf{u}_k$ are taken into account when computing the matrix $\mathbf{Y}_{t,N}$. Thus, in the rest of this article, it is assumed that in (1) $\mathbf{u}_k = 0$ for all $k \geq 0$. In addition, to further simplify the notation, the case for $t = N-1$ is considered, meaning that our goal is to obtain $\mathbf{x}_0$, and thus the matrices $\mathbf{Y}_{t,N}, \mathbf{E}_{t,N}$, and $\Phi_N(\mathbf{x})$ are denoted as $\mathbf{Y}, \mathbf{E}$, and $\Phi(\mathbf{x})$, respectively.

Consider $\mathbf{x}_0$, the state of the plant at $k = 0$, and the system's evolution for N steps as specified in (1) (for $\mathbf{u}_k = 0$) for some attack vectors $\mathbf{e}_0,...,\mathbf{e}_{N-1}$ applied via sensors from set $\mathcal{K} = \{i_1,...,i_q\} \subseteq \mathcal{S}$, where $|\mathcal{K}| \leq q_{\max}$ and the corresponding matrix $\mathbf{E} = [\mathbf{e}_0|\mathbf{e}_1|\ldots|\mathbf{e}_{N-1}]$. Furthermore, consider the case where $|\mathbf{w}_k| \preceq \epsilon_{w_k} \in \mathbb{R}^p$ and $|\mathbf{v}_k| \preceq \epsilon_{v_k} \in \mathbb{R}^n$, $k = 0,1,...,N-1$, that is, the process and element noise vectors are element-wise bounded. Define

$$\mathbf{Y}_{\mathbf{w},\mathbf{v}} = [\mathbf{y}_0|\mathbf{y}_1|\ldots|\mathbf{y}_{N-1}].$$

Note that the matrix $\mathbf{Y}_{\mathbf{w},\mathbf{v}}$ contains measurements of the system including noise. Finally, $\bar{\mathbf{Y}} = [\bar{\mathbf{y}}_0|\bar{\mathbf{y}}_1...|\bar{\mathbf{y}}_{N-1}]$ denotes the sensor measurements (plant outputs) that would be obtained in this case if the system were noiseless, that is, for $\|\epsilon_{w_k}\|_2 = \|\epsilon_{v_k}\|_2 = 0$ (meaning that $\bar{\mathbf{y}}_k = \mathbf{C}\mathbf{A}^k\mathbf{x}_0 + \mathbf{e}_k, k = 0,1,...,N-1$).

Now, consider the following optimization problem

$$\begin{aligned} P_0(\mathbf{Y}): \quad & \min_{\mathbf{E},\mathbf{x}} \|\mathbf{E}\|_{l_0} \\ \text{subject to} \quad & \mathbf{E} = \mathbf{Y} - \Phi(\mathbf{x}). \end{aligned} \tag{12}$$

As previously described,

$$(\mathbf{x}_0, \mathbf{E}) = \arg\max P_0(\bar{\mathbf{Y}}), \tag{13}$$

where $q = \|\mathbf{E}\|_{l_0} \leq q_{\max}$. However, the "ideal" (noiseless) measurements from $\bar{\mathbf{Y}}$ are not available to the estimator; the estimator can only use the measurements in the matrix $\mathbf{Y}_{\mathbf{w},\mathbf{v}}$. In addition, it is worth noting that $(\mathbf{x}_0, E)$ may not even be a feasible point for problem $P_0(\mathbf{Y}_{\mathbf{w},\mathbf{v}})$ when there are noisy measurements. Consequently, problem $P_0(\mathbf{Y})$ should be adapted to handle nonideal models that capture noise and modeling errors.

To achieve this, consider the following problem that relaxes the equality constraint from (12) by including a noise allowance

$$P_{0,\Delta}(\mathbf{Y}): \quad \min_{\mathbf{E},\mathbf{x}} \|\mathbf{E}\|_{l_0}$$
$$\text{subject to } |\mathbf{Y} - \Phi(\mathbf{x}) - \mathbf{E}| \preceq \Delta. \tag{14}$$

Here, the matrix $\Delta \in \mathbb{R}^{p\times N}$ contains nonnegative tolerances $\delta_{j,i}$ for each sensor i, $i = 1,...,p$, in each of the N steps j, that is, $\Delta = [\delta_0 | \delta_1 | \dots | \delta_{N-1}]$, $\delta_i \in \mathbb{R}^p, i = 0,1,...,N-1$. The solution to (14) is

$$(\mathbf{x}_{0,\Delta}, \mathbf{E}_\Delta) = \arg\max P_{0,\Delta}(\mathbf{Y}_{\mathbf{w},\mathbf{v}}),$$
$$q_\Delta = \|\mathbf{E}_\Delta\|_{l_0}. \tag{15}$$

Note that $P_{0,\mathbf{0}^{p\times N}}(\mathbf{Y}) = P_0(\mathbf{Y})$, for all $\mathbf{Y} \in \mathbb{R}^{p\times N}$.

To allow for the use of (14) as an attack-resilient state estimator, it is necessary to ensure that $P_{0,\Delta}(\mathbf{Y})$ has a feasible point $(\mathbf{x}, \mathbf{E})$ such that $\|\mathbf{E}\|_{l_0} \le q_{\max}$; this condition has to be satisfied for all $\mathbf{Y} \in \mathbb{R}^{p\times N}$ that could be "generated" by the system when at most $q_{\max}$ sensors have been attacked, which can be guaranteed with an appropriate initialization of the matrix Δ. From (1), it follows that for $k = 0,1,...,N-1$

$$\mathbf{y}_k = \mathbf{C}\mathbf{A}^k\mathbf{x}_0 + \mathbf{e}_k + \mathbf{C}\sum_{i=0}^{k-1}\mathbf{A}^{k-1-i}\mathbf{v}_i + \mathbf{w}_k$$
$$= \bar{\mathbf{y}}_k + \mathbf{C}\sum_{i=0}^{k-1}\mathbf{A}^{k-1-i}\mathbf{v}_i + \mathbf{w}_k.$$

If $|(\mathbf{A}^{k-1-i})|$ is used to denote the matrix whose elements are absolute values of the corresponding elements of the matrix $\mathbf{A}^{k-1-i}$, then

$$|\mathbf{y}_k - \bar{\mathbf{y}}_k| \preceq |\mathbf{C}|\sum_{i=0}^{k-1}|(\mathbf{A}^{k-1-i})|\mathbf{v}_i| + |\mathbf{w}_k|$$
$$\preceq |\mathbf{C}|\sum_{i=0}^{k-1}|(\mathbf{A}^{k-1-i})|\epsilon_{v_i} + \epsilon_{w_k}$$
$$= \bar{\delta}_k. \tag{16}$$

Therefore, for $\delta_k \succeq \bar{\delta}_k$ $(k = 0,...,N-1)$, it follows that $(\mathbf{x}_0, \mathbf{E})$ from (13) is a feasible point for the problem $P_{0,\Delta}(\mathbf{Y}_{\mathbf{w},\mathbf{v}})$, meaning that there exists $(\mathbf{x}_{0,\Delta}, E_\Delta)$ from (15) such that $q_\Delta = q \le q_{\max}$. Hence, the solution of $P_{0,\Delta}(\mathbf{Y}_{\mathbf{w},\mathbf{v}})$ from (14) can be used as a state estimator in the sense that if at most $q_{\max}$ sensors have been compromised it would provide a solution where the size of row-support of $\mathbf{E}_\Delta$ is not larger than $q_{\max}$.

ROBUSTNESS OF $P_{0,\Delta}(\mathbf{Y})$ STATE ESTIMATION

To perform robustness analysis for the $P_{0,\Delta}(\mathbf{Y})$ optimization problem, it is assumed that the matrix Δ satisfies the aforementioned conditions. Consider $(\mathbf{x}_{0,\Delta}, \mathbf{E}_\Delta)$ from (15) and a matrix $\Sigma \in \mathbb{R}^{p\times N}$ such that

$$\mathbf{Y} - \Phi(\mathbf{x}_{0,\Delta}) - \mathbf{E}_\Delta = \Sigma. \tag{17}$$

Here, $|\Sigma| \preceq \Delta$. In addition, because $(\mathbf{x}_0, \mathbf{E})$ is a feasible point for $P_{0,\Delta}(\mathbf{Y})$, it follows that

$$q = \|\mathbf{E}\|_{l_0} \ge \|\mathbf{E}_\Delta\|_{l_0} = q_\Delta,$$

implying that $\|\mathbf{E} - \mathbf{E}_\Delta\|_{l_0} \le 2q$. The goal is to provide a bound on $\|\Delta\mathbf{x}\|_2$, where

$$\Delta\mathbf{x} = \mathbf{x}_{0,\Delta} - \mathbf{x}_0. \tag{18}$$

If $\Delta\mathbf{E}$ is defined as $\Delta\mathbf{E} = \mathbf{E}_\Delta - \mathbf{E}$, it holds that

$$\Delta\mathbf{E} = (\mathbf{Y}_{\mathbf{w},\mathbf{v}} - \Phi(\mathbf{x}_{0,\Delta}) - \Sigma) - (\overline{\mathbf{Y}} - \Phi(\mathbf{x}_0))$$
$$= \underbrace{(\mathbf{Y}_{\mathbf{w},\mathbf{v}} - \overline{\mathbf{Y}} - \Sigma)}_{\Delta\mathbf{Y}} - \Phi(\Delta\mathbf{x}_0).$$

Denote by $\Delta\mathbf{y}_0,...,\Delta\mathbf{y}_{N-1}$ the columns of the matrix $\Delta\mathbf{Y}$ (that is, $\Delta\mathbf{Y} = [\Delta\mathbf{y}_0,...,\Delta\mathbf{y}_{N-1}]$). From (16) and (17) it follows that

$$|\Delta\mathbf{y}_k| \preceq \bar{\delta}_k + \delta_k \preceq 2\delta_k.$$

Accordingly, a bound on $\|\Delta\mathbf{x}\|_2$, is

$$\max_{\Delta\mathbf{x}} \|\Delta\mathbf{x}\|_2 \tag{19}$$
$$\|\Phi(\Delta x) - \Omega\|_{l_0} \le 2q, \tag{20}$$
$$\Omega \preceq 2\Delta. \tag{21}$$

Since $q \le q_{\max}$, the feasible space can be increased by relaxing constraint (20) to

$$\|\Delta\mathbf{Y} - \Phi(\Delta\mathbf{x})\|_{l_0} \le 2q_{\max}. \tag{22}$$

Therefore, the goal is to bound $\Delta\mathbf{x}$ for which there exists $\Omega \in \mathbb{R}^{p\times N}$ that satisfies (21), and for where *at least* $p - 2q_{\max}$ rows of the matrix $\Phi(\Delta\mathbf{x}) - \Omega$ are zero rows. Let F and $\mathcal{K}_F \subset \mathcal{S}$ to denote the number of rows $\Phi\Delta(\mathbf{x})$ that are zero rows and the set of corresponding sensors, respectively. This means that at least $F_1 = p - 2q_{\max} - F$ rows of $\Phi(\Delta\mathbf{x})$ are equal to the rows of Ω, which are nonzero, and let $\mathcal{K}_{F_1} \subset \mathcal{S}$ to denote sensors corresponding to those rows. It is worth noting here that $|\mathcal{K}_F \cup \mathcal{K}_{F_1}| = p - 2q_{\max}$ and $\mathcal{K}_F \cap \mathcal{K}_{F_1} = \emptyset$.

Since $\mathcal{K}_F \subset \mathcal{S}$ contains indices of zero rows of $\Phi(\Delta\mathbf{x})$, it follows that $\mathbf{O}_{\mathcal{K}_F}\Delta\mathbf{x} = 0$, where $\mathbf{O}_{\mathcal{K}_F}$ is defined as in (4). In addition, $\mathbf{O}_{\mathcal{K}_{F_1}}\Delta\mathbf{x} = \Omega_{\mathcal{K}_{F_1}}$, where for $\Omega = [\omega_1 | \omega_2 | \dots | \omega_N]$ (that is, $\omega_i, i = 1,...N$ are columns of Ω such that $|\omega_i| \preceq 2\delta_i$), and

$$\Omega_{\mathcal{K}_{F_1}} = \begin{bmatrix} P_{\mathcal{K}_{F_1}}\omega_1 \\ P_{\mathcal{K}_{F_1}}\omega_2 \\ \vdots \\ P_{\mathcal{K}_{F_1}}\omega_N \end{bmatrix} \text{ and } \Delta_{\mathcal{K}_{F_1}} = \begin{bmatrix} P_{\mathcal{K}_{F_1}}\delta_1 \\ P_{\mathcal{K}_{F_1}}\delta_2 \\ \vdots \\ P_{\mathcal{K}_{F_1}}\delta_N \end{bmatrix}.$$

Consequently, for $\Delta\mathbf{x}$ to satisfy constraints (22) and (21), there have to exist sets $\mathcal{K}_F, \mathcal{K}_{F_1} \subset \mathcal{S}$ such that

$$|\mathcal{K}_F| = F, \ |\mathcal{K}_{F_1}| = p - 2q_{\max} - F, \tag{23}$$
$$\mathcal{K}_F \cap \mathcal{K}_{F_1} = \emptyset, \tag{24}$$
$$\mathbf{O}_{\mathcal{K}_F}\Delta\mathbf{x} = 0, \tag{25}$$
$$|\mathbf{O}_{\mathcal{K}_{F_1}}\Delta\mathbf{x}| \preceq 2\Delta_{\mathcal{K}_{F_1}}. \tag{26}$$

Now, consider the polyhedron $\mathbb{P}$ defined with constraints (23)–(26). From its definition it follows that the point $\Delta\mathbf{x} = 0$ belongs to the polyhedron. In addition, the polyhedron $\mathbb{P}$ is bounded. To show this, start with the following lemma.

Lemma 2

For any two sets $\mathcal{K}_F, \mathcal{K}_{F_1} \subset \mathcal{S}$ such that $|\mathcal{K}_F| = F$, $|\mathcal{K}_{F_1}| = p - 2q_{\max} - F$ and $\mathcal{K}_F \cap \mathcal{K}_{F_1} = \emptyset$,

$$\operatorname{rank}(\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = n. \tag{27}$$

Proof

From [18], $q_{\max} = \lfloor s/2 - 1 \rfloor$, where s is the cardinality of the smallest set $\mathcal{K} \subseteq \mathcal{S}$ for which the matrix $\mathbf{O}_{\mathcal{K}^{\mathbb{C}}}$ has nontrivial kernel. Note that $|\mathcal{K}^{\mathbb{C}}| = p - s$, and since $s \geq 2q_{\max} + 1 > 2q_{\max}$, it follows that $|\mathcal{K}^{\mathbb{C}}| < p - 2q_{\max}$. Now consider any set $\mathcal{K}_1$ for which $|\mathcal{K}_1^{\mathbb{C}}| \geq p - 2q_{\max}$, meaning that $|\mathcal{K}_1| \leq 2q_{\max} < s$. Thus, $\mathbf{O}_{\mathcal{K}_1^{\mathbb{C}}}$ does not have nontrivial kernel (since $\mathcal{K}$ is the smallest such matrix), meaning that columns of $\mathbf{O}_{\mathcal{K}_1^{\mathbb{C}}}$ are linearly independent. Thus, since $\mathbf{O}_{\mathcal{K}_1^{\mathbb{C}}} \in \mathbb{R}^{N|\mathcal{K}_1^{\mathbb{C}}| \times n}$, it follows that $\operatorname{rank}(\mathbf{O}_{\mathcal{K}_1^{\mathbb{C}}}) = n$. This is satisfied for any $\mathcal{K}_1^{\mathbb{C}}$ with at least $p - 2q_{\max}$ sensors, and hence (27) holds since the set $\mathcal{K}_F \cup \mathcal{K}_{F_1}$ contains $p - 2q_{\max}$ sensors. ■

Theorem 1

The polyhedron $\mathbb{P}$ defined by constraints (23)–(26) is bounded.

Proof

Assume the opposite, namely, that $\mathbb{P}$ is unbounded. Then there exist a feasible point $\Delta\mathbf{x} \in \mathbb{P}$ and a direction $\mathbf{d} \in \mathbb{R}^n$ such that $\mathbf{d} \neq 0$ and for any $\epsilon > 0$, $\Delta\mathbf{x} + \epsilon\mathbf{d} \in \mathbb{P}$ [32]. Therefore, $\mathbf{O}_{\mathcal{K}_F}(\Delta\mathbf{x} + \epsilon\mathbf{d}) = 0$, and, since $\Delta\mathbf{x} \in \mathbb{P}$, it follows that $\mathbf{O}_{\mathcal{K}_F}\mathbf{d} = 0$. In addition,

$$|\mathbf{O}_{\mathcal{K}_{F_1}}(\Delta\mathbf{x} + \epsilon\mathbf{d})| \preceq 2\Delta_{\mathcal{K}_{F_1}} \tag{28}$$

implies that $\mathbf{O}_{\mathcal{K}_{F_1}}\mathbf{d} = \mathbf{0}$ (otherwise for any nonzero element of the vector $\mathbf{O}_{\mathcal{K}_{F_1}}\mathbf{d}$, when $\epsilon \to \infty$ the absolute value of that element in vector $\epsilon\mathbf{O}_{\mathcal{K}_{F_1}}\mathbf{d}$ will be unbounded and the constraint (28) will be violated). Therefore, $\mathbf{d}$ belongs to the kernel of $\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$, that is, $\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}\mathbf{d} = \mathbf{0}$. However, from Lemma 2, $\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$ has full rank (i.e., $\operatorname{rank}(\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = n$), meaning that it has a nontrivial kernel and thus $\mathbf{d} = 0$, which violates the initial assumption and concludes the proof. ■

As a direct consequence of Theorem 1, it follows that maximal $\|\Delta\mathbf{x}\|_2$ is bounded, and the attacker cannot use modeling errors and the corresponding relaxation of the l_0 optimization problem to introduce an unbounded error in the attack-resilient state estimator.

Bounding the State-Estimation Error

Theorem 1 allows bounding $\|\Delta\mathbf{x}\|_2$, the error of the resilient state estimator $P_{\Delta,0}(\mathbf{Y}_{\mathbf{w},\mathbf{v}})$, by noticing that the maximal value of a convex function over a polyhedron can be obtained at a vertex of the polyhedron [33]. Thus, to determine the maximal $\|\Delta\mathbf{x}\|_2$ over the polyhedron $\mathbb{P}$ it is sufficient to compute $\|\Delta\mathbf{x}\|_2$ at each vertex of the polyhedron. The vertices of the polyhedron satisfy

$$\underbrace{\begin{bmatrix}\mathbf{O}_{\mathcal{K}_F}\\ \mathbf{O}_{\mathcal{K}_{F_1}}\end{bmatrix}}_{\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}} \cdot \Delta\mathbf{x} = \begin{bmatrix}0\\ 2\Delta^{+-}_{\mathcal{K}_{F_1}}\end{bmatrix}, \tag{29}$$

where $\Delta^{+-}_{\mathcal{K}_{F_1}}$ denotes a vector such that $|\Delta^{+-}_{\mathcal{K}_{F_1}}| = \Delta_{\mathcal{K}_{F_1}}$ (that is, with elements whose absolute values are equal to the corresponding elements of $\Delta_{\mathcal{K}_{F_1}}$). It is worth noting that there are $2^{|\mathcal{K}_{F_1}| \cdot N}$ such elements and thus $2^{|\mathcal{K}_{F_1}| \cdot N}$ vertices of the polyhedron. Finally, since $\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$ is a full-rank matrix ($\operatorname{rank}(\tilde{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = \operatorname{rank}(O_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = n$), vertex points can be found as

$$\begin{aligned}\Delta\mathbf{x}_{\text{ver}} &= (\tilde{\mathbf{O}}^T_{\mathcal{K}_F \cup \mathcal{K}_{F_1}} \tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}})^{-1} \tilde{\mathbf{O}}^T_{\mathcal{K}_F \cup \mathcal{K}_{F_1}} \begin{bmatrix}0\\ 2\Delta^{+-}_{\mathcal{K}_{F_1}}\end{bmatrix}\\ &= \tilde{\mathbf{O}}^{\dagger}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}} \begin{bmatrix}0\\ 2\Delta^{+-}_{\mathcal{K}_{F_1}}\end{bmatrix},\end{aligned} \tag{30}$$

where $\tilde{\mathbf{O}}^{\dagger}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$ denotes the pseudoinverse of matrix $\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$. Consequently, for any sets $\mathcal{K}_F$ and $\mathcal{K}_{F_1}$ that satisfy (23) and (24), by checking all $2^{|\mathcal{K}_{F_1}| \cdot N}$ vertices defined by (30), the maximal $\|\Delta\mathbf{x}\|_2$ can be determined for the corresponding polyhedron. However, since

$$\|\Delta\mathbf{x}_{\text{ver}}(\Delta^{+-}_{\mathcal{K}_{F_1}})\|_2 = \|\Delta\mathbf{x}_{\text{ver}}(-\Delta^{+-}_{\mathcal{K}_{F_1}})\|_2,$$

where $\Delta\mathbf{x}_{\text{ver}}(\Delta^{+-}_{\mathcal{K}_{F_1}})$ denotes the solution of (30) for specific $\Delta^{+-}_{\mathcal{K}_{F_1}}$, it is only needed to evaluate norms at $2^{|\mathcal{K}_{F_1}| \cdot N - 1}$ points (that is, vertices). Furthermore, to provide a bound on $\|\Delta\mathbf{x}\|_2$ for all $\Delta\mathbf{x}$ that satisfy (21) and (22), all such sets $\mathcal{K}_F$ and $\mathcal{K}_{F_1}$ have to be considered. Therefore, it is necessary to evaluate all possible values for F. From the definition, $F \geq 0$. On the other hand, from (25), $\mathcal{K}_F$ has a nontrivial kernel, meaning that, as in the proof of Lemma 2, $F = |\mathcal{K}_F| \leq p - s \leq p - 2q_{\max} - 1$. Finally, from (30) the bound can be over approximated as

$$\|\Delta\mathbf{x}\|_2 \leq 2 \max_{F, F_1} \lambda^{\max}_{\tilde{\mathbf{O}}^{\dagger}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}} \|\Delta_{\mathcal{K}_{F_1}}\|_2 = 2 \max_{F, F_1} \frac{\|\Delta_{\mathcal{K}_{F_1}}\|_2}{\lambda^{\min}_{\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}}}, \tag{31}$$

where $\lambda^{\max}_{\tilde{\mathbf{O}}^{\dagger}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}}$ denotes the maximal singular value of the matrix $\tilde{\mathbf{O}}^{\dagger}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$, while $\lambda^{\min}_{\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}}$ denotes the smallest singular value of the matrix $\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$ (and this is nonzero since it is a full-rank matrix).

Note that the matrix Δ captures several sources of modeling errors (for example, noise, jitter, or synchronization errors). Since (31) is linear in Δ, the estimation error bound obtained by evaluating the $\|\Delta\mathbf{x}\|_2$ at vertices of the polyhedron will be less than or equal to the sum of the estimation error bounds computed separately for each error component. Therefore, it is

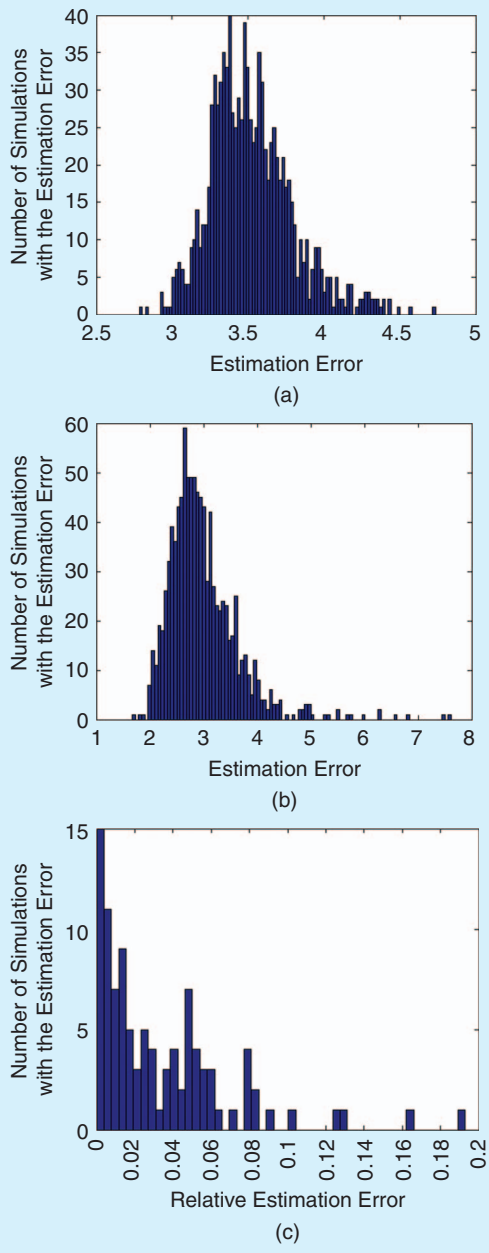


FIGURE 2 Simulation results for 1000 runs of 100 randomly selected systems with $n = 10$ states and $p = 5$ sensors. (a) State-estimation error, (b) state-estimation error, and (c) relative state-estimation error. Histograms of (a) a system with the obtained error bound equal to 41.43, (b) a system with the obtained error bound equal to 35.74, and (c) the maximal relative state-estimation error for all 100 systems.

possible to separately analyze the impact for each source of modeling error on the robustness of the state estimator.

To obtain the bound from (30), in the general case, the number of times that (30) has to be solved is

$$\sum_{F=0}^{p-s} \binom{p}{F} \binom{p-F}{p-2q_{\max}-F} 2^{(p-2q_{\max}-F)N-1}.$$

Note that, for almost all systems, meaning that for *almost all* pairs of matrices $\mathbf{A} \times \mathbf{C} \in \mathbb{R}^{n\times n} \times \mathbb{R}^{p\times n}$ (that is, the set of matrices for which the property does not hold has Lebesgue measure zero), the number of correctable errors using the previous $N = n$ measurement vectors is (maximal and) equal to $q_{\max} = \lceil p/2 - 1 \rceil$ [18]; in this case, $s = p$, and thus F can only take the value zero, meaning that the error needs to be evaluated in $p2^{n-1}$ if p is an odd number, or $(p(p-1)/2)2^{2n-1}$ if the system has an even number of sensors. This effectively limits the above-described exhaustive search for systems with a large number of states or sensors. In this case, it is possible to use a more conservative bound introduced in [34] that significantly reduces the complexity of the procedure used for the computation.

EVALUATION

To evaluate the conservativeness of the error bound described in the previous section, two types of systems are considered, systems with $n = 10$ states and $p = 5$ sensors and those with $n = 20$ states and $p = 11$ sensors. For each system type, 100 systems were generated with measurement models satisfying the requirements that the rows of the C matrix have unit magnitude and the Δ matrices had elements between zero and two. In addition, for each of the 200 systems, the state-estimation error $\Delta\mathbf{x} = \| \mathbf{x}_{0,\Delta} - \mathbf{x}_0 \|_2$ was evaluated in 1000 experiments for various attack and noise realizations. Attack and noise profiles were chosen randomly assuming a uniform distribution of 1) the number of attacked sensors between zero and two for systems with five sensors and between zero and five for systems with 11 sensors, 2) attack vectors on the compromised sensors between –10 and 10, chosen independently for each attacked sensor, and 3) noise realizations between the noise bounds specified by matrices Δ.

The considered case was with the window size N equal to the number of system states (that is, $N = n$). Comparison between the bounds computed as described in the previous section and simulation results are shown in Figures 2 and 3. Figures 2(a), (b), and 3(a) present histograms of $\| \Delta\mathbf{x} \|_2$ errors for all 1000 scenarios for three randomly selected systems. As can be seen, the computed bound is an order of magnitude larger than the average state-estimation error for each system. However, for each system $\mathfrak{S}$, more relevant is the ratio between the worst-case observed state-estimation error for all 1000 simulations, that is, $\max_{i=1:1000} \| \Delta\mathbf{x}_{\mathfrak{S}} \|_2$, and the computed error bound $\text{MAX\_}\| \Delta\mathbf{x}_{\mathfrak{S}} \|_2$ for the system. Thus, the relative estimation error is considered, defined for each system $\mathfrak{S}$ as

$$\text{Rel\_error}_{\mathfrak{S}} = \frac{\max_{i=1:1000} \Delta\mathbf{x}_{\mathfrak{S}}}{\text{MAX\_}\| \Delta\mathbf{x}_{\mathfrak{S}} \|_2}.$$

Histograms of the relative errors for both types of systems are presented in Figures 2(c) and 3(b). For the systems with $n = 10$ states, the maximal relative error reaches almost 20% of computed bounds, while for larger systems (with $n = 20$ states), the maximal relative error is 2% of computed bounds.

Conservativeness of the presented results is (at least partially) caused by the fact that for each system only random initial points were considered and random uncorrelated attack vectors and noise profiles/modeling errors. Thus, the errors obtained through simulation do not represent the worst-case errors. For each system, to obtain scenarios that result in the worst-case estimation errors, it is necessary to derive the corresponding attack vector (and the initial state), which is beyond the scope of this article. This is illustrated in histograms of relative estimation errors for systems with different sizes. As in the histograms from Figures 2(c) and 3(b), a decrease in the obtained maximal relative estimation error was observed in simulations, with an increase in the system size n (and thus increase in the window size $N = n$). One of the reasons is that with the increase of N, the number of attack vectors also increases, and due to the random-actor selection of the vectors, probabilities to incorporate a worst-case attack are reduced.

On the other hand, for systems with fewer states (like $n = 1, 2,$ or 3) we were able to generate initial states and attack vectors for which the computed bounds are tight, that is, the error $\|\Delta \mathbf{x}\|_2$ is equal to the obtained bounds. For these attacks, it was assumed that the attacker, which controlled up to $q_{\max}$ sensors, had full knowledge of the system state and the measurements of noncompromised sensors. The attacker's goal was to maximize the state-estimation error when the proposed attack-resilient state estimation error is used.

CASE STUDY: ATTACK-RESILIENT CRUISE CONTROL ON AUTONOMOUS GROUND VEHICLE

In this section, the use of the presented development framework is illustrated on a design of secure cruise control of the LandShark vehicle [35], a fully electric unmanned ground vehicle shown in Figure 4(a). In a tethered mode, the robot can be fully teleoperated from the operator control unit. However, in this scenario, the operator only specifies the desired vehicle speed, while the onboard control has to ensure that all of the safety requirements are satisfied even if some of the sensors are under attack.

Vehicle Modeling

To obtain a dynamical model of the vehicle, the standard differential-drive vehicle model can be used [Figure 4(b)] [36]. Here, F_l and F_r denote forces on the left and right set of wheels, respectively, and B_r is the mechanical resistance of the wheels to rolling. The vehicle position is specified by its x and y coordinates, θ denotes the heading angle of the vehicle measured from the x axis, while v is the speed of the vehicle in this direction. The LandShark employs skid steering, meaning that to make a turn it is necessary to generate enough torque to overcome the sticking force S_l. Therefore, when $(B/2)|F_l - F_r| \geq S_l$ the wheels start to slide sideways (that is, the vehicle begins to turn). Consequently, if it is assumed that the wheels do not slip, the dynamical model of the vehicle is

$$\begin{aligned}
\dot{v} &= \begin{cases} \frac{1}{m}(F_l + F_r - (B_s + B_r)v), & \text{if turning,} \\ \frac{1}{m}(F_l + F_r - B_r v), & \text{if not turning,} \end{cases} \\
\dot{\omega} &= \begin{cases} \frac{1}{J_t}\left(\frac{B}{2}(F_l - F_r) - B_l \omega\right), & \text{if turning,} \\ 0, & \text{if not turning,} \end{cases} \\
\dot{\theta} &= \omega, \quad \dot{x} = v\sin(\theta), \quad \dot{y} = v\cos(\theta).
\end{aligned}$$

Also, $\omega = 0$ if the vehicle is not turning.

Finally, to estimate the state of the vehicle for cruise control (that is, its speed and position), three sensors are employed, two speed encoders, one on each side, and a GPS. The GPS provides time-stamped global position and speed, and the rotation angle can be obtained from

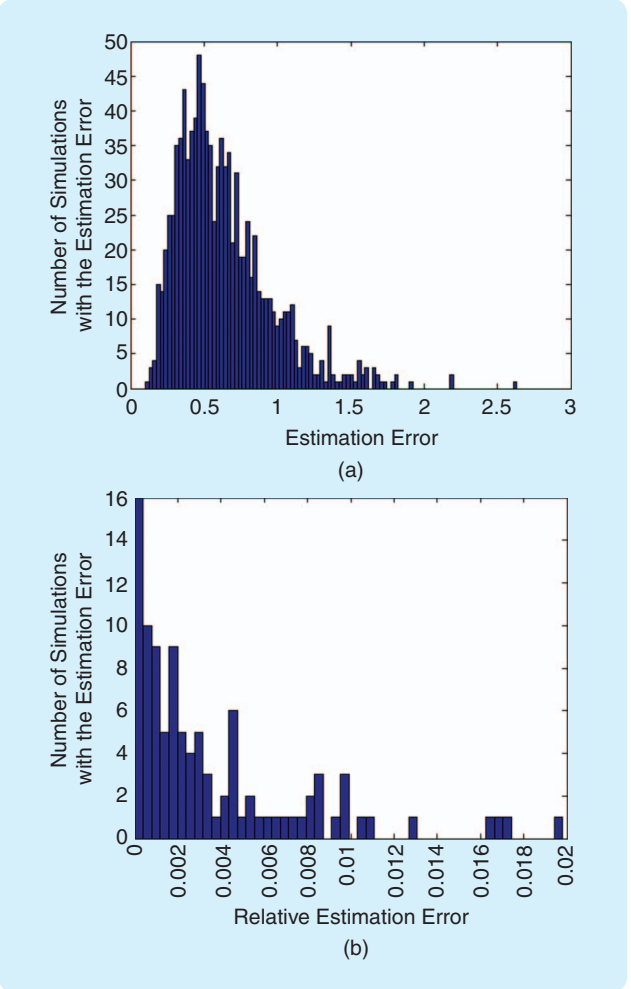


FIGURE 3 Simulation results for 1000 runs of 100 randomly selected systems with $n = 20$ states and $p = 11$ sensors. (a) State-estimation error and (b) relative state-estimation error. Histograms for (a) a system with the obtained error bound equal to 155.98 and (b) the maximal relative state-estimation error for all 100 systems.

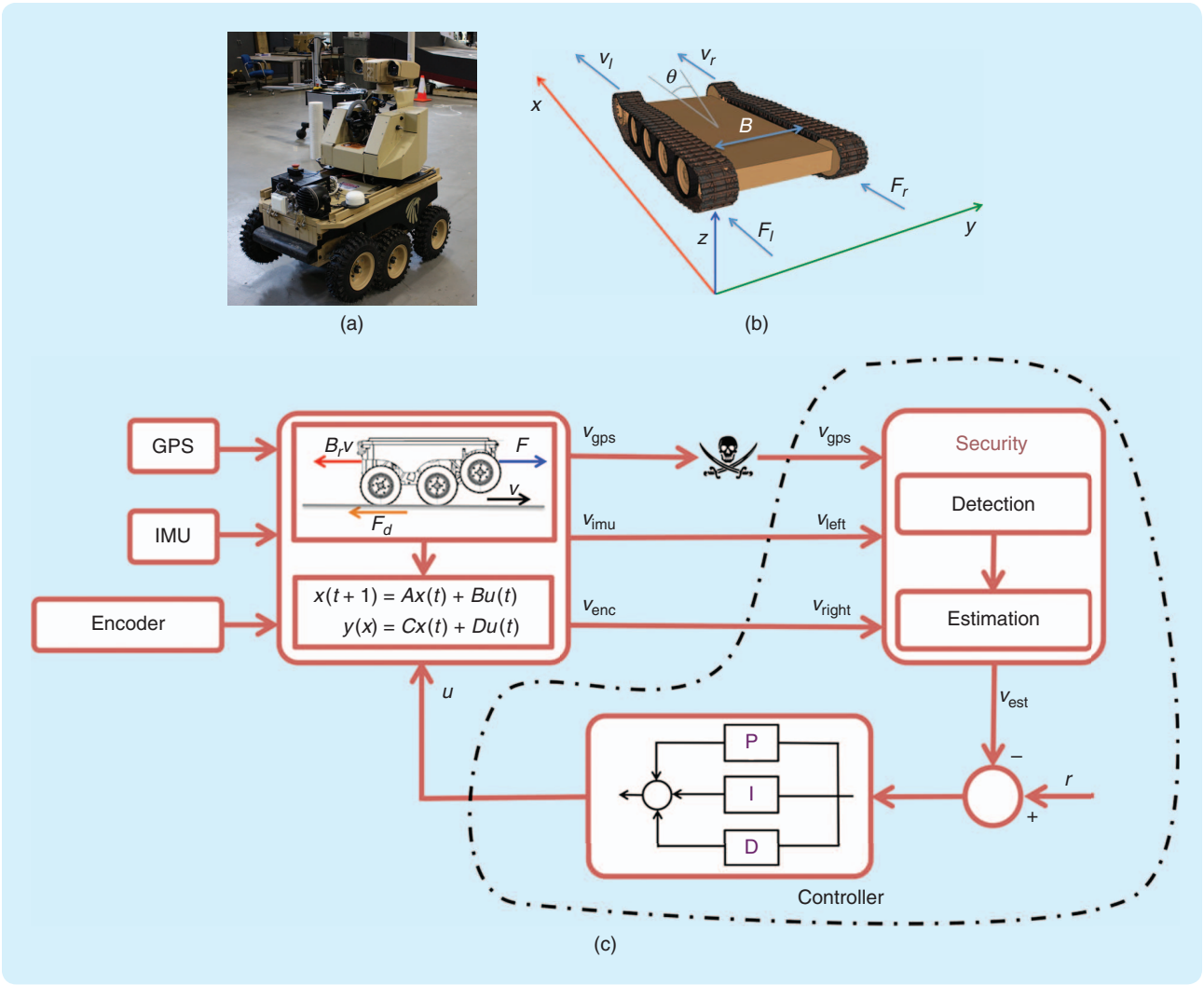


FIGURE 4 The LandShark unmanned ground vehicle. (a) The vehicle, (b) the coordinate system and variables used to derive the model, and (c) the control system diagram used for cruise control. IMU: intertial measurement unit.

the encoders (which can be translated into rotational velocity and finally into linear velocity). Note that other sensors can be used to estimate the state of the vehicle, for instance, linear acceleration measurements obtained from an inertial measurement unit (IMU), or visual odometry estimates computed by optical flow algorithms from a camera feed. However, to illustrate the use (and robustness) of the attack-resilient state estimator, only the encoders and GPS are employed.

The above model presents a high-level one of the vehicle, describing only the motion equations. The forces F_l and F_r, which can be considered as inputs to the model, are derived from the vehicle's electromotors and are affected by the motors, gearbox, and wheels. Thus, a six-state linear model of this low-level electromechanical system based on the model from [36] was derived, which is then used to obtain a local state (that is, velocity) feedback controller that provides the desired F_l, F_r levels.

System Architecture

All sensors on the LandShark vehicle are connected to the CPU, which implements the state estimator and controller through independent serial buses, while the motors are connected to the CPU via motor drivers [as presented in Figure 4(c)]. Since the speed of the vehicle is bounded, the attack-resilient state-estimator from (14) can be formulated as a mixed-integer linear programming problem

$$\begin{aligned} &\min_{\gamma,\mathbf{E},\mathbf{x}} \quad 1_p^T \gamma \\ -\delta_k \preceq \mathbf{y}_k \quad &-\mathbf{C}\mathbf{A}^k\mathbf{x} - \mathbf{e}_k \preceq \delta_k, \ k = 0, \ldots, N-1, \\ -\gamma_j \alpha \cdot 1_N' \preceq \ &\mathbf{E}_j' \preceq \gamma_j \alpha \cdot 1_N', \ j = 1, \ldots, p, \end{aligned}$$

where $\mathbf{E}_j'$ and $\mathbf{e}_k$ denote the jth row and kth column of the matrix $E \in \mathbb{R}^{p \times N}$, respectively. Here, $\gamma = [\gamma_1, \ldots, \gamma_p] \in \{0,1\}^p$ are binary optimization variables representing, for each sensor j, whether the sensor is considered *attacked* ($\gamma_j = 1$)

The controller, written in C++ language, subscribes to each sensor measurement through the master and sends inputs to the motor driver to maintain the desired cruise speed.

or *safe* ($\gamma_j = 0$), and α is a sufficiently large positive constant. Note that since the robot cannot obtain a speed larger than 20 mi/h, all sensor measurements larger than the value have to be obtained from compromised sensors and thus can be discarded. Hence, it can be assumed that elements of attack vectors cannot be larger than the maximal speed.

The developed resilient controller is executed on top of Linux OS and the Robot Operating System (ROS) middleware [37]. ROS is a meta-operating system that facilitates the development of robotic applications using a publish/subscribe mechanism in which a master superintends every operation. Associated with each sensor is a driver that takes care of getting time-stamped information from the sensor and publishing this data in the ROS format to the ROS master. The controller, written in C++ language, subscribes to each sensor measurement (called topics) through the master and sends inputs to the motor driver to maintain the desired cruise speed. The tool ROSLab [38] was used to describe the architecture of the control system.

Experiments

Figure 5 presents a deployment of the robot during experiments run on a uneven tiled surface and an uneven grassy field. From the developed graphical user interface, it is demonstrated that the robot can reach and maintain the desired reference speed even when one of the sensors is under attack, as shown in Figure 6. Figure 6(a) presents speed estimates from the encoders and GPS. Each of the sensors was attacked at some point, with attacks such that their measurements would result in the speed estimate equal to 4 m/s, except in the last period of the simulation when the experiment was switched to an alternating attack on the left encoder.

However, as shown in Figure 6(b), when the attack-resilient controller is active, the robot reaches and maintains the desired speed of 1 m/s. On the other hand, if the state estimator is disabled and instead a simple observer is employed (as in the interval between 68 s and 73 s, the highlighted area in Figure 6), even when one of the sensors is under attack the robot cannot reach the desired state (for example, it can even be forced to stop). Videos of the LandShark experiments can be seen at [39].

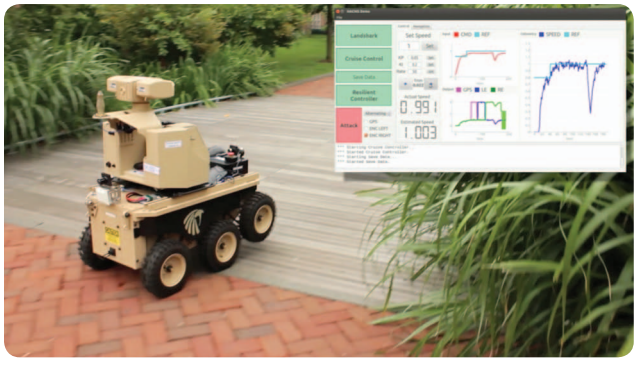


FIGURE 5 The deployment of the LandShark on a tiled pathway. The inset picture displays the user interface used in experiments.

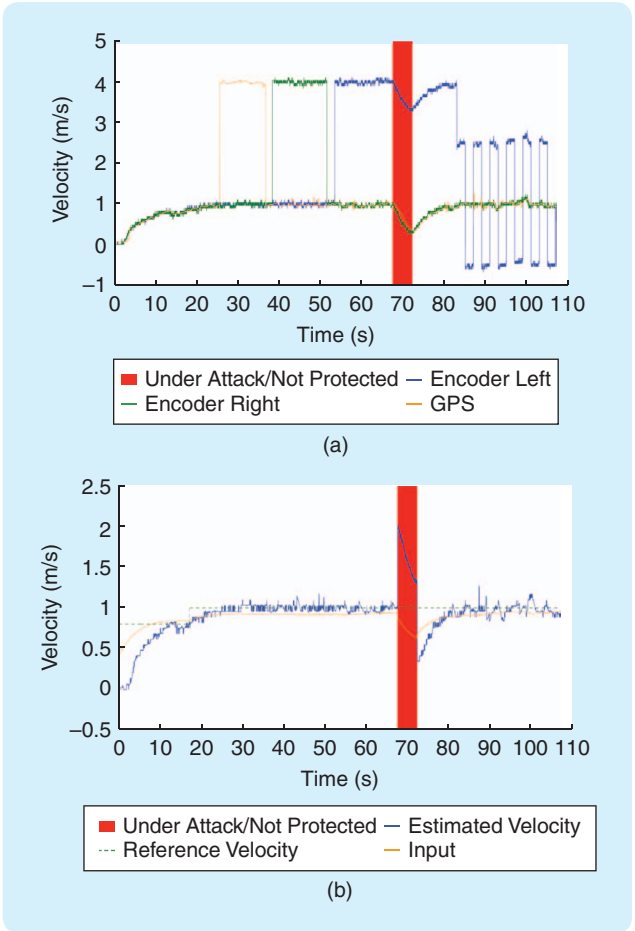


FIGURE 6 Experimental results: (a) cruise control under attack and (b) estimated and reference velocity versus input. (a) A comparison of velocity estimated from the encoders' and GPS measurements and (b) reference speed, the estimated speed, and the input applied to the motors.

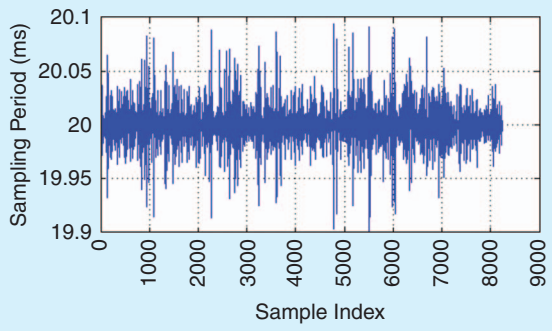


FIGURE 7 Times between consecutive left encoder measurements.

Robustness Analysis

All ROS nodes are executed in the *run-to-completion* manner. Thus, although the execution period for the controller node is 20 ms, other instantiated nodes might affect its execution (that is, the controller might execute with a variable period). Each sensor has its own clock, and all measurements are time-stamped before being transmitted to the controller. Yet, since relative changes in obtained measurements are used, time-synchronization error between sensors does not accumulate. In addition, there is a huge discrepancy between sensors' sampling jitters. For example, encoders' sampling jitters are bounded by 100 μs (as shown in Figure 7), while the GPS has highly variable jitter with maximal measured values up to 125 ms. Therefore, it is not possible to use the idealized discrete-time model from (9), but rather the full input compensation has to be done as in (7) and (8), before the state estimator is executed.

Consequently, a bound on GPS error is determined from manufacturer specifications, worst-case sampling jitter, and synchronization error and is experimentally validated to be $\delta_{k,1} \leq 0.4$ m/s. On the other hand, each encoder has 192 cycles per revolution, resulting in a measurement error of 0.5%. Thus, since the maximal achievable vehicle speed is 20 m/s, it follows that for both encoders $\delta_{k,2} = \delta_{k,2} \leq 0.1$ m/s.

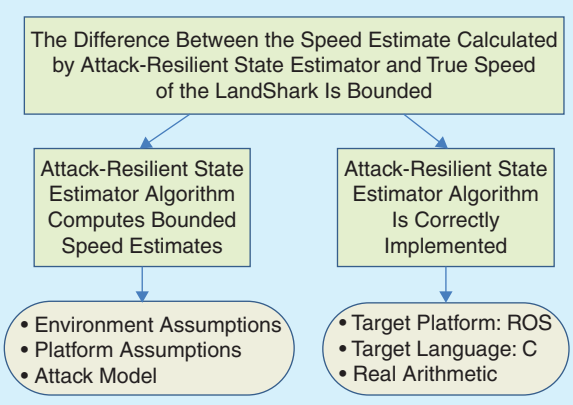


FIGURE 8 Top-level claims of the assurance case.

For these values, the computed state-estimation error bound is 0.72 m/s. Note that the conservativeness of the bound is mostly caused by the large worst-case GPS sampling jitter.

ASSURANCE CASE FOR THE RESILIENT CRUISE-CONTROL IMPLEMENTATION

In a complex CPS design project, when a large team is engaged in design and validation and verification activities, it can be difficult to maintain a centralized, coherent view of the system and its associated evidence in all its detail. Assurance cases have been proposed as a means to organize the evidence into a coherent argument that captures what evidence is available, what assumptions have been made in the design process, and how each piece of evidence contributes to the overall assurance. For the considered case study, a detailed assurance case was constructed, covering both a mathematical model of the state estimator and its physical environment as well as a software implementation of the controller. The goal has been to gain understanding of what levels of modeling are involved in the design and implementation of a resilient control system, what reasoning techniques are used at each level, and what assumptions are likely to be made at each level of abstraction, as well as how these assumptions can be justified by guarantees established in a lower-level model. In this article, an overview of the developed assurance case is presented, focusing on the implementation guarantees. The detailed assurance case description can be found in [40].

In a straightforward generalization from [41], an assurance case can be defined as *a documented body of evidence that provides a convincing and valid argument that a system has desired critical properties for a given application in a given environment*. A common example of such a critical property is system safety, even in the presence of attacks, in which case the argument is known as a safety case. The top-level claims of the assurance case are shown in Figure 8, and the argument is partitioned into two parts. One part is concerned with the *algorithmic* correctness of the state estimator and the tracking proportional-integral derivative (PID) controller. This part of the assurance case can be referred to as the control-level argument as it deals with mathematical models of the estimator and relies on the robustness analysis presented in the previous sections. The other part addresses the implementation of the overall controller and the way it is deployed on the LandShark platform. The argument also specifies assumptions and the implementation context. The assurance case relies on three categories of assumptions.

- Attack assumptions represent a model of the attacker capabilities. Attacks on sensor data are considered, without any restrictions on the attacker's capability to manipulate a stream of sensor data. However, our assumption is that fewer than half of the sensors are attacked. Thus, given that the LandShark platform has three sensors, at most one sensor can be compromised at any time. There is no direct way to prove that this

assumption holds since it describes the limitation on the capability of the attacker. Indirect justification for the attack model can be derived from the implementation of the control system. In particular, sensors are implemented as different ROS nodes and publish their readings on separate ROS topics, making it more difficult for an attacker to compromise multiple sensor streams.

- Environmental assumptions describe the intended operating environment of the vehicle, which are used to derive a model of its dynamics.
- Platform assumptions and the implementation context deal with the properties of the LandShark platform, including a certain sampling frequency, expected latency of sensing and actuation, and maximum actuation jitter, which have been validated on the platform as shown in the previous section.

In general, when an assurance case for the whole vehicle is constructed, these platform assumptions correspond to claims made in other parts of the assurance case.

Implementation-level Assurance Arguments

This part of the argument is presented in Figure 9. The strategy is to separate the argument into two subclaims. The first subclaim covers the platform-independent implementation of the attack-resilient state-estimator algorithm and PID controller, implemented as a *step function* periodically invoked by the platform. The second subclaim considers the deployment of the step function within a platform-specific wrapper that handles periodic invocation of the step function and its connection to the streams of sensor data and makes speed estimates available to other modules in the system. Arguments for both subclaims are instances of the model-manipulation strategy. The step function is obtained using Simulink Coder and has been verified using the methods introduced in [42] and [43]. The wrapper for the step function is produced from the architectural model of the LandShark platform, which captures ROS topics and their respective publishers and subscribers. The wrapper generator has been implemented in Coq [44] and supplies a proof that a) the wrapper subscribes to the sensor topics as specified in the architectural model, b) subscribed values are passed to the parameters of the step function, and c) the step function is invoked with the period specified in the architectural model. This proof is used as evidence for the technique subclaim, and a review of the architectural model is performed as evidence for the model subclaim.

DISCUSSION AND FUTURE WORK

In this article, methods to provide performance guarantees in CPSs in the presence of sensor attacks have been presented. By focusing on the design of attack-resilient cruise control for autonomous ground vehicles, control-theoretic challenges in attack-resilient state estimation for dynamical systems with noise and modeling errors have been described. Also, an l_0-norm-based state estimator has been introduced along with an algorithm to derive a bound for the state-estimation error caused by noise and modeling errors in the presence of attacks. Furthermore, methods to map control requirements to specifications imposed on the underlying execution platform have been presented. Finally, an approach to construct an assurance case for the considered system has been described. This overall assurance case is the subject of an on-going multi-institutional project funded by the DARPA High-Assurance Cyber Military Systems program. Some of the platform assumptions made in the argument have been claims delivered by other parts of the overall assurance case.

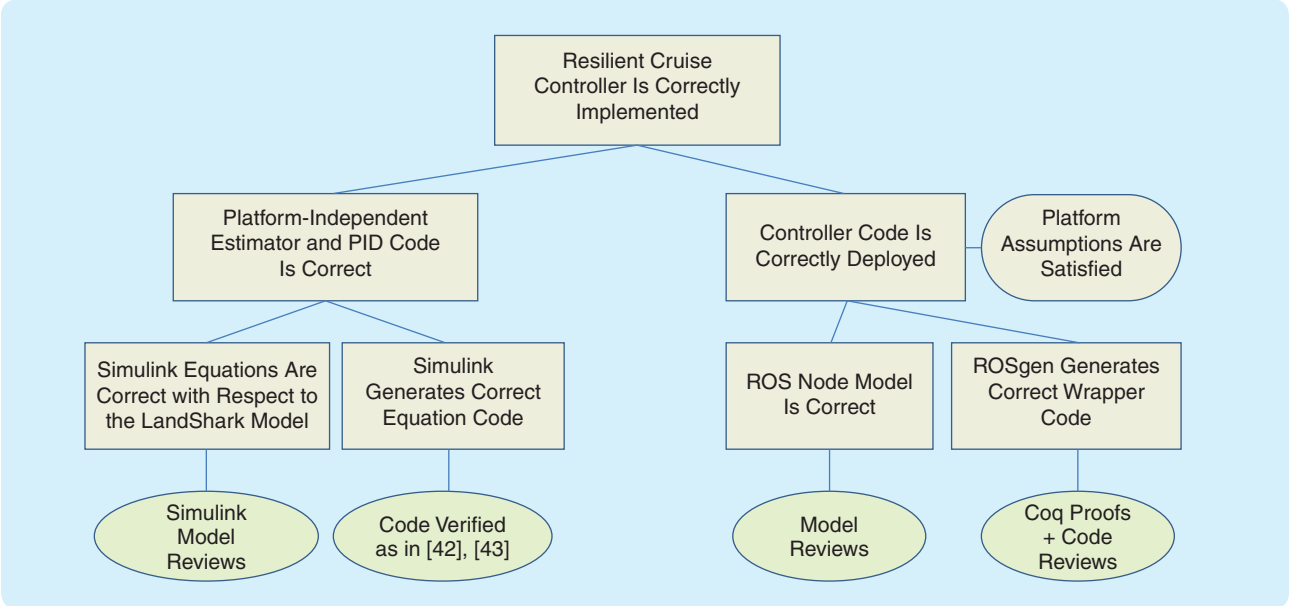


FIGURE 9 An argument for the code-level claims. ROS: Robot Operating System; PID: proportional, integral, derivative.

Note that, during the control-design phase for resilient CPSs, designers usually face limitations of the platform since a certain degree of redundancy in the control loop is needed to achieve the necessary detection and mitigation capabilities. Sensor redundancy is (relatively) easy to handle by adding an additional sensor payload to the platform, such as odometers, IMUs, and GPSs in the LandShark case study. On the other hand, sensor redundancy is only useful if the attacker is not able to compromise all (or more than $q_{\max}$) of the available sensors, which could be violated if the attacker gets access to the local network used to communicate the measurements. However, the biggest limitation is the redundancy of actuators. For example, if actuators on one side of the vehicle are compromised, the skid-steer approach used in LandShark is not feasible. Furthermore, a synthesis of control-task code and proof of its correctness relies on the guarantees provided by the platform services. Therefore, in some cases, the assumption needed to make the proofs go through may turn out to be too restrictive for the platform operating system.

Note that the proposed attack-resilient state-estimation algorithm, while providing accuracy guarantees, does not guarantee attack detection and identification of compromised sensors due to the presence of noise and modeling errors. An avenue for future work would be to provide a sound attack-identification procedure. In addition, the presented estimator requires solving combinatorial optimization problems in each iteration. Therefore, it would be beneficial to derive computationally more efficient methods for attack-resilient state estimation that would potentially provide relaxed performance guarantees.

ACKNOWLEDGMENTS

This material is based on research sponsored by DARPA under agreement number FA8750-12-2-0247. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government. This work was also supported in part by NSF CNS-1505701, CNS-1505799 grants, and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy. Preliminary versions of some of these results have been presented in [25], [34], and [40].

AUTHOR INFORMATION

Miroslav Pajic (miroslav.pajic@duke.edu) received the Dipl. Ing. and M.S. degrees in electrical engineering from the University of Belgrade, Serbia, in 2003 and 2007, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, in 2010 and 2012, respectively. He is currently an assistant professor in the Department of Electrical and Computer Engineering at Duke University. He also holds a secondary appointment in the Department of Computer Science. His research interests focus on the design and analysis of cyberphysical systems and, in particular, real-time and embedded systems, distributed/networked control systems, and high-confidence medical devices and systems. He can be contacted at Duke University, 100 Science Dr., Hudson Hall, Room 130, Durham, NC 27708 USA.

James Weimer is a research assistant professor at the University of Pennsylvania in the Department of Computer and Information Science. He received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University and was previously a postdoctoral researcher in the Department of Automatic Control at the Royal Institute of Technology KTH, Stockholm. His research interests focus on the design and analysis of closed-loop and data-driven cyberphysical systems with application to medicine and security.

Nicola Bezzo is an assistant professor in systems and information engineering at the University of Virginia. He also holds a secondary appointment in electrical and computer engineering and is a member of the LINK Lab. He previously was a postdoctoral researcher with the PRECISE Center at the University of Pennsylvania. He received the B.S. and M.S. degrees in electrical engineering from the Politecnico di Milano, Italy, in 2006 and 2008, respectively, and the Ph.D degree in electrical and computer engineering from the University of New Mexico in 2012. His research interests include motion planning of unmanned aerial and ground robotic vehicles under uncertainties, heterogeneous robotic systems, attack-resilient control of cyberphysical systems, and codesign and rapid prototyping of mobile robotic systems.

Oleg Sokolsky is a research associate professor of computer and information science at the University of Pennsylvania. His research interests include the application of formal methods to the development of cyberphysical systems, architecture modeling and analysis, specification-based monitoring, as well as software-safety certification. He received the Ph.D. degree in computer science from Stony Brook University, New York.

George J. Pappas received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, in 1998. He is currently the Joseph Moore Professor and chair of Electrical and Systems Engineering at the University of Pennsylvania. He also holds secondary appointments in computer and information sciences and mechanical engineering and applied mechanics. He is a member of the GRASP Lab and the PRECISE Center. He currently serves as the deputy dean for research in the School of Engineering and Applied Science. His current research interests include hybrid systems and control, embedded control systems, cyberphysical systems, hierarchical and distributed control systems, and networked control systems, with applications to robotics, unmanned aerial vehicles, biomolecular networks, and green buildings.

Insup Lee is the Cecilia Fitler Moore Professor of Computer and Information Science and Director of the PRECISE Center at the University of Pennsylvania. He also holds a

secondary appointment in the Department of Electrical and Systems Engineering. He received the B.S. degree in mathematics from the University of North Carolina, Chapel Hill, and the Ph.D. degree in computer science from the University of Wisconsin, Madison. His research interests include cyberphysical systems, real-time embedded systems, formal methods and tools, high-confidence medical device systems, and software engineering. The theme of his research activities has been to assure and improve the correctness, safety, and timeliness of life-critical embedded systems.

REFERENCES

[1] J. Slay and M. Miller, "Lessons learned from the Maroochy water breach," in *Proc. Critical Infrastructure Protection*, 2007, pp. 73–82.

[2] R. Langner, "StuxNet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[3] T. M. Chen and S. Abu-Nimeh, "Lessons from StuxNet," *Computer*, vol. 44, no. 4, pp. 91–93, 2011.

[4] R. M. Lee, M. J. Assante, and T. Conway. (2014, Dec.). German steel mill cyber attack. *Industrial Control Systems* [Online]. Available: https://ics.sans.org/media/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf

[5] K. Zetter. (2015). A cyberattack has caused confirmed physical damage for the second time ever. [Online]. Available: https://www.wired.com/2015/01/german-steel-mill-hack-destruction

[6] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *Proc. 2010 IEEE Symp. Security and Privacy*, 2010, pp. 447–462.

[7] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Security,* 2011.

[8] A. Greenberg. (2015). Hackers remotely kill a jeep on the highway. [Online]. Available: http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[9] G. Jaffe and T. Erdbrink, (2011. Dec. 4). Iran says it downed U.S. stealth drone; Pentagon acknowledges aircraft downing. *Washington Post*. [Online]. Available: https://www.washingtonpost.com/world/national-security/iran-says-it-downed-us-stealth-drone-pentagon-acknowledges-aircraft-downing/2011/12/04/gIQAyxa8TO_story.html?utm_term=.d19fc66c0faf

[10] S. Peterson and P. Faramarzi, (2011. Dec. 15). Iran hijacked U.S. drone, says Iranian engineer. *Christian Science Monitor.* [Online]. Available: http://www.csmonitor.com/World/Middle-East/2011/1215/Exclusive-Iran-hijacked-US-drone-says-Iranian-engineer

[11] D. Shepard, J. Bhatti, and T. Humphreys. (2012. Aug. 1). Drone hack: Spoofing attack demonstration on a civilian unmanned aerial vehicle. *GPS World*. [Online]. Available: http://gpsworld.com/drone-hack/

[12] J. S. Warner, and R. G. Johnston, "A simple demonstration that the Global Positioning System (GPS) is vulnerable to spoofing," *J. Security Admin.*, vol. 25, no. 2, pp. 19–27, 2002.

[13] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *Proc. 18th ACM Conf. Computer and Communications Security*, 2011, pp. 75–86.

[14] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *Cryptographic Hardware and Embedded Systems-CHES 2013*. New York: Springer-Verlag, 2013, pp. 55–72.

[15] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems," in *Proc. 1st Int. Conf. High Confidence Networked Systems*, 2012, pp. 55–64.

[16] R. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," in *Proc. Int. Federation Automatic Control World Congress*, 2011, pp. 90–95.

[17] F. Pasqualetti, F. Dorfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Automat. Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[18] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Trans. Automat. Control*, vol. 59, no. 6, pp. 1454–1467, 2014.

[19] S. Sundaram, M. Pajic, C. Hadjicostis, R. Mangharam, and G. Pappas, "The wireless control network: Monitoring for malicious behavior," in *Proc. 49th IEEE Conf. Decision and Control*, 2010, pp. 5979–5984.

[20] F. Miao, M. Pajic, and G. Pappas, "Stochastic game approach for replay attack detection," in *Proc. 52nd. IEEE Conf. Decision and Control*, 2013, pp. 1854–1859.

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

[22] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Syst. Mag.*, vol. 35, no. 1, pp. 93–109, 2015.

[23] Y. Mo, T. H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proc. IEEE*, vol. 100, no. 1, pp. 195–209, 2012.

[24] C. Kwon, W. Liu, and I. Hwang, "Security analysis for cyber-physical systems against stealthy deception attacks," in *Proc. IEEE American Control Conf.*, 2013, pp. 3344–3349.

[25] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. Pappas, "Robustness of attack-resilient state estimators," in *Proc. ACM/IEEE Int. Conf. Cyber-Physical Systems*, 2014, pp. 163–174.

[26] Y. Shoukry, A. Puggelli, P. Nuzzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, "Sound and complete state estimation for linear dynamical systems under sensor attacks using Satisfiability Modulo Theory solving," in *Proc. American Control Conf.*, 2015, pp. 3818–3823.

[27] Y. Shoukry, P. Nuzzo, N. Bezzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada. (2015). A satisfiability modulo theory approach to secure state reconstruction in differentially flat systems under sensor attacks. [Online]. Available: https://arxiv.org/abs/1509.03262

[28] P. Antsaklis and A. Michel, *Linear Systems*. New York: McGraw Hill, 1997.

[29] Y. Shoukry and P. Tabuada. (2013). Event-triggered state observers for sparse sensor noise/attacks. [Online]. https://arxiv.org/abs/1309.3511

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

[31] W. Zhang, M. Branicky, and S. Phillips, "Stability of networked control systems," *IEEE Control Syst. Mag.*, vol. 21, no. 1, pp. 84–99, 2001.

[32] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, 1st ed. Nashua, NH: Athena Scientific, 1997.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[34] M. Pajic, P. Tabuada, I. Lee, and G. Pappas, "Attack-resilient state estimation in the presence of noise," in *Proc. 54th IEEE Annu. Conf. Decision and Control*, Dec. 2015, pp. 5827–5832.

[35] Black-I Robotics LandShark UGV. [Online]. Available: http://www.blackirobotics.com/LandShark_UGV_UC0M.html

[36] J. J. Nutaro, *Building Software for Simulation: Theory and Algorithms, with Applications in C++*. New York: Wiley, 2010.

[37] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. Open-Source Software Workshop Int. Conf. Robotics and Automation*, 2009.

[38] N. Bezzo, J. Park, A. King, P. Gebhard, R. Ivanov, and I. Lee, "Demo abstract: Roslab—A modular programming environment for robotic applications," in *Proc. ACM/IEEE Int. Conf. Cyber-Physical Systems*, 2014, p. 214.

[39] [Online]. Available: http://people.duke.edu/mp275/research/CPS_security.html

[40] J. Weimer, O. Sokolsky, N. Bezzo, and I. Lee, "Towards assurance cases for resilient control systems," in *Proc. IEEE Int. Conf. Cyber-Physical Systems, Networks, and Applications*, 2014, pp. 1–6.

[41] Adelard, *ASCAD–The Adelard Safety Case Development (ASCAD) Manual*. 1998.

[42] M. Pajic, J. Park, I. Lee, G. J. Pappas, and O. Sokolsky, "Automatic verification of linear controller software," in *Proc. 12th Int. Conf. Embedded Software*, 2015, pp. 217–226.

[43] J. Park, M. Pajic, I. Lee, and O. Sokolsky, "Scalable verification of linear controller software," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS),* M. Chechik and J.F. Raskin, Eds. New York: Springer-Verlag, 2016, pp. 662–679.

[44] The Coq Development Team. (2004). *The Coq Proof Assistant Reference Manual*, LogiCal Project, version 8.0 [Online]. Available: http://coq.inria.fr