4-2017

Design and Implementation of Attack-Resilient Cyber-Physical Systems

Miroslav Pajic

James Weimer
University of Pennsylvania, weimerj@cis.upenn.edu

Nicola Bezzo
University of Pennsylvania

Oleg Sokolsky
University of Pennsylvania, sokolsky@cis.upenn.edu

George Pappas
University of Pennsylvania, pappasg@cis.upenn.edu

See next page for additional authors

Recommended Citation

Miroslav Pajic, James Weimer, Nicola Bezzo, Oleg Sokolsky, George Pappas, and Insup Lee, "Design and Implementation of Attack-Resilient Cyber-Physical Systems", *IEEE Control Systems* 37(2), 66-81. April 2017. http://dx.doi.org/10.1109/MCS.2016.2643239

IEEE Control Systems (Volume: 37, Issue: 2, April 2017) https://ieeexplore.ieee.org/document/7879899/

Design and Implementation of Attack-Resilient Cyber-Physical Systems

Abstract

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 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.

Keywords

Security, Actuators, State estimation, Linear systems, Resilience, Real-time systems, cyber-physical systems, safety-critical software, security of data, attack-resilient cyberphysical systems, attack-resilient state estimators, security, high-profile attacks, critical infrastructure, Maroochy Water breach, industrial systems, StuxNet virus attack, industrial supervisory control and data acquisition system, German steel mill cyberattack, high-assurance military systems, attack vulnerability, RQ-170 Sentinel US drone, CPS, safety-critical control systems, networked control systems, embedded control systems

Disciplines

Computer Engineering | Computer Sciences

Comments

IEEE Control Systems (Volume: 37, Issue: 2, April 2017) https://ieeexplore.ieee.org/document/7879899/

Author(s)

Miroslav Pajic, James Weimer, Nicola Bezzo, Oleg Sokolsky, George Pappas, and Insup Lee

Design and Implementation of Attack-Resilient Cyber-Physical Systems

Miroslav Pajic, James Weimer, Nicola Bezzo, Oleg Sokolsky
George J. Pappas, 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 (e.g., the StuxNet virus attack on an industrial SCADA system [2], [3] and the German Steel Mill cyber attack [4], [5]), to attacks on modern vehicles [6], [7], [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 US drone [9], [10], [11]. These incidents have seriously raised security awareness in Cyber-Physical Systems (CPS), which feature tight coupling of computation and 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 that present a complex interaction of a large number of embedded Electronic Control Units (ECUs), 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; as shown in [6], [7], [8], using simple methods an attacker can disrupt the operation of a car, even taking complete control over it.

In general, attacks on a cyber-physical system may affect all of its components – computational nodes and communication networks are subject to intrusions, and 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 non-invasively via the physical environment. For instance, non-invasive attacks on GPS-based navigation systems [12], [13], [14], and anti-lock braking systems [15] 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 do not alone guarantee resilience of control systems to attacks on physical components of the system. On the other hand, getting access to an internal network would allow the attacker to compromise sensors$\rightarrow$controller$\rightarrow$actuators communication; from the control perspective these attacks can also be modeled as additional adversary signals introduced via the sensors and actuators [16]. 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 as the primary function of CPS, where the attacker can (1) take over a sensor and supply wrong or untimely 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 have attracted a lot of attention, with several efforts focused on the use of control-level techniques, which exploit a model of the 'normal' system behavior, for attack-detection and identification in CPS (e.g., [17], [16], [18], [19], [20], [21], [22], [23]). For instance, methods for attack-detection based on the use of standard residual probability based detectors were presented in [24], [25], [22], [23], while the problem of state estimation in the presence of sensors attacks was addressed in [18], [19], [26], [27].

By contrast, attacks on the execution platform prevent the correct operation of the control system as in the cases where the attacker can disrupt execution of control tasks. Defense against such 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, and 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 (e.g., attack-detection) and the desired control performance, as long as the worst-case execution time of the control task is, for example, 20 milliseconds and output jitter is no more than 2 milliseconds.

Consequently, for attack-resilient control in CPS 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 OS and utilized networking, and facilitate reasoning about attack-resilience across different implementation layers.

In this article, we describe our efforts on the development of attack-resilient CPS. Specifically, a case study is considered – design of a resilient cruise controller for an autonomous ground vehicle, focusing on one component of the system, namely attack-resilient state estimator (RSE) and the performance guarantees in the presence of attacks. Hence, the article starts by addressing the problem of attack-resilient state estimation, before providing robustness guarantees for the implemented RSE (building on our work from [26]). 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, it is described how implementation issues such as jitter, latency and synchronization errors can be mapped into parameters of the state estimation procedure. This effectively enables mapping control performance requirements into real-time (i.e., timing related) specifications imposed on the underlying platform. Finally, it is presented how to construct an assurance case for the system that covers both a mathematical model of the state estimator and its physical environment, 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.

I. 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. This has been motivated by the fact that the same controllers can be used as in the case without attacks, if the controller is able to reasonably well estimate the state of the controlled physical process even if some of the sensor measurements and actuator commands have been compromised. For deterministic (i.e., noiseless) linear time-invariant systems, the correct state estimate in the presence of sensor attacks can be obtained as the solution of l_0 optimization problems [18], [19]. In addition, in [27], [28], the authors presented estimation techniques for linear and differentially-flat systems, respectively, based on the use of Satisfiability Modulo Theories (SMT) 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. This, as discussed in the introduction, limits their applicability in real systems since it is unclear what level of resiliency guarantees they could provide with more realistic sensing, actuation, and execution models. Hence, the focus of this section is on the attack-resilient state estimation for dynamical systems with bounded noise and modeling errors, and 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, before the estimator and the procedure to bound its worst-case estimation error in the presence of attacks is introduced.

1) Notation and Terminology: In this article, the following notation is used. For a set $\mathcal{S}$, $|\mathcal{S}|$ denotes the cardinality (i.e., size) of the set, while for two sets $\mathcal{S}$ and $\mathcal{R}$, $\mathcal{S} \setminus \mathcal{R}$ is used to denote the set of elements in $\mathcal{S}$ that are not in $\mathcal{R}$. In addition, for a set $\mathcal{K} \subset \mathcal{S}$, $\mathcal{K}^{\complement}$ specifies the complement set of $\mathcal{K}$ with respect to $\mathcal{S}$ – i.e., $\mathcal{K}^{\complement} = \mathcal{S} \setminus \mathcal{K}$. Also, $\mathbb{R}$ is used to denote the set of reals, and $\mathbf{1}'_N$ to denote the row vector of size N containing all ones. Finally, for any sequence of $\alpha_i,\ i \geq 0$, since the sum $\sum_0^{-1} \alpha_i$ contains no elements, to simplify the notation it is assumed that it is equal to zero – i.e., $\sum_0^{-1} \alpha_i = 0$.

Furthermore, $\mathbf{A}^T$ is used to indicate the transpose of matrix $\mathbf{A}$, while i^{th} 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

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

while l_0 norm of vector $\mathbf{e}$ is the size of supp($\mathbf{e}$) – i.e., $\|\mathbf{e}\|_{l_0} = |\text{supp}(\mathbf{e})|$. Note that, although l_0 is not formally a norm, in this article we will abuse the terminology and referred to it as a norm in order to maintain consistency with the terminology used in previous work on this topic (e.g., [19]).

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

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

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

A. System Model

In this article, a Linear-Time Invariant (LTI) system is considered, specified as

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

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 $\mathcal{K}^C = \mathcal{S} \setminus \mathcal{K}$, 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 (e.g., 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 additional vector e^a_k is added to the plant input at each step $k \geq 0$. As shown in [19], 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 both subsets of the plant's sensors and actuators are compromised. Consequently, the analysis and results presented in this article 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 (i.e., enabling the resilient state-estimation), even with accurate estimates of the plant's state system stability can not 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 [17].

1) Attack-resilient State Estimation for Noiseless Dynamical Systems: For linear systems without noise (i.e., systems from (1) where $\mathbf{w}_k = \mathbf{0}$ and $\mathbf{v}_k = \mathbf{0}$, for all $k \geq 0$), a l_0-norm based method to extract state estimate in presence of attacks is introduced in [19]. To obtain the plant's state at any time-step t (i.e., $\mathbf{x}_t$), the proposed procedure utilizes 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

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

Here, $\mathbf{Y}_{t,N} = [\tilde{\mathbf{y}}_{t-N+1}|\tilde{\mathbf{y}}_{t-N+2}|\dots|\tilde{\mathbf{y}}_t] \in \mathbb{R}^{p \times N}$ 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, $\Phi_N : \mathbb{R}^n \to \mathbb{R}^{p\times N}$ is a linear mapping defined as $\Phi_N(\mathbf{x}) = \left[\mathbf{C}\mathbf{x}|\mathbf{C}\mathbf{A}\mathbf{x}|\ldots|\mathbf{C}\mathbf{A}^{N-1}\mathbf{x}\right]$, which captures the system's evolution over N steps caused by the initial state $\mathbf{x}$.

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

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

The critical observation here is that for a noiseless LTI system there is a pattern of zeros (i.e., zero-rows) in the matrix $\mathbf{E}_{t,N}$ that corresponds to the non-attacked 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 [19], 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 state after N steps when up to q sensors are under attack if and only if for all $\mathbf{x} \in \mathbb{R}\setminus\{\mathbf{0}\}$,

$$|\text{supp}(\mathbf{C}\mathbf{x}) \cup \text{supp}(\mathbf{C}\mathbf{A}\mathbf{x}) \cup \ldots \cup \text{supp}(\mathbf{C}\mathbf{A}^{N-1}\mathbf{x})| > 2q.$$

In this work, q_{max} is used to denote 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 utilized 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 that $q \leq q_{max}$, 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 Cayley-Hamilton theorem [29] it cannot be further increased when more than n previous measurements are used – i.e., q_{max} obtains the maximal value for $N = n$. Finally, beside the measurement window size N, q_{max} only depends on the system's dynamics (i.e., matrices $\mathbf{A}$ and $\mathbf{C}$), as was characterized in [30], [19]. To formally capture this dependency, consider the following notation – for any set $\mathcal{K} = \{k_1, ..., k_{|\mathcal{K}|}\} \subseteq S$, where $k_1 < k_2 < ... < k_{|\mathcal{K}|}$, the matrices $\mathbf{O}_\mathcal{K}$ and $P_\mathcal{K}$ are defined as

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

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 1 in its j^{th} position.

Definition 1 ([30]): An LTI system with the form as in (1) is said to be s-sparse observable if for every set $\mathcal{K} \subset \mathcal{S}$ of size s (i.e., $|\mathcal{K}| = s$), the pair $(\mathbf{A}, P_{\mathcal{K}^\complement}\mathbf{C})$ is observable.

From the results in [30], [19], the following lemma holds.

Lemma 1: q_{max} is equal to the maximal s for which the system is $2s$-sparse observable.

2) 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 uncertainty of parameters estimation during the system modeling; 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 non-idealities of control system implementation and limitations of the utilized 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 discrete-time model (1) of the system. Consequently, to be able to deal with continuous-time plants it is necessary 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 $\mathbf{A}_c, \mathbf{B}_c, \mathbf{C}_c$ are of the appropriate dimensions.

First, consider setups where all plant's output are sampled (i.e., 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 Fig. 1. Here, the k^{th} 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 Fig. 1(b). Using the approach from [31], [32], 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 [29]

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

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

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

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, inter-sampling 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 utilize the 'ideal' discrete-time model of the system of the form (1) where for all $k \geq 0$, $T_{s,k} = T_s$ and $\tau_k = 0$

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

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

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

Finally, from the equation above 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 (i.e., $T_{s,k} - T_s$) and latency (i.e., 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.

a) Effects of Synchronization Errors: To simplify the presentation, only systems where the sensors do not have a common clock source are considered – i.e., 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 j^{th} row of $\mathbf{C}$, meaning that the synchronization error introduces a measurement error defined as

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

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 (e.g., 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 $\mathbf{v}_k^{syn} \in \mathbb{R}^p$ describing modeling errors caused by synchronization errors between sensors.

B. 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 – i.e., 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 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 subsection, 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) $\mathbf{u}_k = \mathbf{0}$ 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 $\mathbf{\Phi}_N(\mathbf{x})$ are denoted as $\mathbf{Y}, \mathbf{E}$ and $\mathbf{\Phi}(\mathbf{x})$, 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 $\mathbf{u}_k = \mathbf{0}$) 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 $|\mathbf{v}_k| \preceq \epsilon_{v_k} \in \mathbb{R}^n$, $k = 0, 1, ..., N-1$ – i.e., the process and element noise vectors are element-wise bounded – and let's define

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

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 was noiseless – i.e., 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

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

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 specified by 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 $P_0(\mathbf{Y}_{\mathbf{w},\mathbf{v}})$ that utilizes noisy sensor measurements. Consequently, there is need to adapt problem $P_0(\mathbf{Y})$ to non-ideal 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

$$\begin{aligned} P_{0,\Delta}(\mathbf{Y}): \qquad & \min_{\mathbf{E},\mathbf{x}} \|\mathbf{E}\|_{l_0} \\ s.\ t. \quad & |\mathbf{Y} - \Phi(\mathbf{x}) - \mathbf{E}| \preceq \mathbf{\Delta}. \end{aligned} \tag{14}$$

Here, the matrix $\mathbf{\Delta} \in \mathbb{R}^{p \times N}$ contains non-negative tolerances $\delta_{j,i}$ for each sensor $i,\ i = 1, ..., p$, in each of the N steps j – i.e., $\mathbf{\Delta} = [\delta_0 | \delta_1 | \dots | \delta_{N-1}]$, $\delta_i \in \mathbb{R}^p, i = 0, 1, ..., N-1$. The solution of the above problem is denoted as

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

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 $\|\mathbf{E}\|_{l_0} \leq q_{max}$; 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. This can be guaranteed with an appropriate initialization of the matrix $\mathbf{\Delta}$. From (1), it follows that for $k = 0, 1, ..., N-1$

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

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

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

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 a solution of the problem – i.e., there exists $(\mathbf{x}_{0,\Delta}, E_\Delta)$ from (15) such that $q_{\mathbf{\Delta}} = q \leq q_{max}$. This means that 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 $\mathbf{E}_{\mathbf{\Delta}}$ is not larger than q_{max}.

C. *Robustness of* $P_{0,\Delta}(\mathbf{Y})$ *State Estimation*

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

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

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

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

implying that $\|\mathbf{E} - \mathbf{E}_{\mathbf{\Delta}}\|_{l_0} \leq 2q$. Our 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 $\Delta\mathbf{E} = \mathbf{E}_{\mathbf{\Delta}} - \mathbf{E}$ it holds that

$$\begin{aligned}\Delta\mathbf{E} &= (\mathbf{Y}_{\mathbf{w},\mathbf{v}} - \Phi(\mathbf{x}_{0,\Delta}) - \mathbf{\Sigma}) - (\bar{\mathbf{Y}} - \Phi(\mathbf{x}_0)) \\ &= \underbrace{(\mathbf{Y}_{\mathbf{w},\mathbf{v}} - \bar{\mathbf{Y}} - \mathbf{\Sigma})}_{\Delta\mathbf{Y}} - \Phi(\Delta\mathbf{x}_0).\end{aligned}$$

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

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

Accordingly, to provide a bound on $\|\Delta\mathbf{x}\|_2$, the following problem can be considered

$$\max_{\Delta\mathbf{x}} \quad \|\Delta\mathbf{x}\|_2 \tag{19}$$

$$\|\Phi(\Delta\mathbf{x}) - \mathbf{\Omega}\|_{l_0} \leq 2q, \tag{20}$$

$$\mathbf{\Omega} \preceq 2\Delta. \tag{21}$$

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

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

Therefore, our goal is to bound $\Delta\mathbf{x}$ for which there exists $\mathbf{\Omega} \in \mathbb{R}^{p\times N}$ that satisfies (21), and for where ***at least*** $p-2q_{max}$ rows of the matrix $\Phi(\Delta\mathbf{x})-\mathbf{\Omega}$ are zero-rows. Lets use 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 $\mathbf{\Omega}$, which are non-zero, and let's use $\mathcal{K}_{F_1} \subset \mathcal{S}$ 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 $\mathbf{O}_{\mathcal{K}_F}\Delta\mathbf{x} = \mathbf{0}$, 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 $\mathbf{\Omega} = [\omega_1|\omega_2|...|\omega_N]$ (i.e., $\omega_i, i = 1,...N$ are columns of $\mathbf{\Omega}$ such that $|\omega_i| \preceq 2\delta_i$), and

$$\mathbf{\Omega}_{\mathcal{K}_{F_1}} = \begin{bmatrix} P_{\mathcal{K}_{F_1}}\omega_1 \\ P_{\mathcal{K}_{F_1}}\omega_2 \\ \vdots \\ P_{\mathcal{K}_{F_1}}\omega_N \end{bmatrix} \quad \mathbf{\Delta}_{\mathcal{K}_{F_1}} = \begin{bmatrix} P_{\mathcal{K}_{F_1}}\delta_1 \\ P_{\mathcal{K}_{F_1}}\delta_2 \\ \vdots \\ P_{\mathcal{K}_{F_1}}\delta_N \end{bmatrix}.$$

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

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

Now, consider the polyhedron $\mathbb{P}$ defined with constraints (23)-(26). From its definition it follows that the point $\Delta\mathbf{x} = \mathbf{0}$ 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 $\mathcal{K}_F \cap \mathcal{K}_{F_1} = \emptyset$,

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

Proof: From [19], $q_{max} = \lceil s/2-1 \rceil$ where s is the cardinality of the smallest set $\mathcal{K} \subseteq \mathcal{S}$ for which the matrix $\mathbf{O}_{\mathcal{K}^\complement}$ has non-trivial kernel. Note that $|\mathcal{K}^\complement| = p-s$, and since $s \geq 2q_{max}+1 > 2q_{max}$, it follows that $|\mathcal{K}^\complement| < p-2q_{max}$. Now consider any set $\mathcal{K}_1$ for which $|\mathcal{K}_1^\complement| \geq p-2q_{max}$, meaning that $|\mathcal{K}_1| \leq 2q_{max} < s$. Thus, $\mathbf{O}_{\mathcal{K}_1^\complement}$ does not have non-trivial kernel (since $\mathcal{K}$ is the smallest such matrix), meaning that columns of $\mathbf{O}_{\mathcal{K}_1^\complement}$ are linearly independent. Thus, since $\mathbf{O}_{\mathcal{K}_1^\complement} \in \mathbb{R}^{N|\mathcal{K}_1^\complement|\times n}$, it follows that $rank(\mathbf{O}_{\mathcal{K}_1^\complement}) = n$. This implies that for any $\mathcal{K}_1^\complement$ 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: Lets assume the opposite, that $\mathbb{P}$ is unbounded; there exist a feasible point $\Delta\mathbf{x} \in \mathbb{P}$ and a direction $\mathbf{d} \in \mathbb{R}^n$ such that $\mathbf{d} \neq \mathbf{0}$ and for any $\epsilon > 0$, $\Delta\mathbf{x} + \epsilon\mathbf{d} \in \mathbb{P}$ [33]. Therefore, $\mathbf{O}_{\mathcal{K}_F}(\Delta\mathbf{x}+\epsilon\mathbf{d}) = \mathbf{0}$, and since $\Delta\mathbf{x} \in \mathbb{P}$ it follows that $\mathbf{O}_{\mathcal{K}_F}\mathbf{d} = \mathbf{0}$. In addition,

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

implies that $\mathbf{O}_{\mathcal{K}_{F_1}}\mathbf{d} = \mathbf{0}$ (otherwise for any non-zero 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 $\mathbf{O}_{\mathcal{K}_F\cup\mathcal{K}_{F_1}}$ – i.e., $\mathbf{O}_{\mathcal{K}_F\cup\mathcal{K}_{F_1}}\mathbf{d} = \mathbf{0}$. However, from Lemma 2, $\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$ has full rank (i.e., rank$(\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = n$), meaning that it has non-trivial kernel and thus $\mathbf{d} = \mathbf{0}$, which violates our initial assumption and concludes the proof. ■

As a direct consequence of the above theorem 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.***

1) Bounding the State-estimation Error: The above theorem allows us to bound $\|\Delta \mathbf{x}\|_2$, 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 in a vertex of the polyhedron [34]. 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 that

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

where $\mathbf{\Delta}^{+-}_{\mathcal{K}_{F_1}}$ denotes a vector such that $|\mathbf{\Delta}^{+-}_{\mathcal{K}_{F_1}}| = \mathbf{\Delta}_{\mathcal{K}_{F_1}}$ (i.e., with elements whose absolute values are equal to the corresponding elements of $\mathbf{\Delta}_{\mathcal{K}_{F_1}}$). 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 (rank$(\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) =$ rank$(\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = n$), vertex points can be found as

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

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 (31), the maximal $\|\Delta \mathbf{x}\|_2$ can be determined for the corresponding polyhedron. However, since

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

where $\Delta \mathbf{x}_{ver}(\mathbf{\Delta}^{+-}_{\mathcal{K}_{F_1}})$ denotes the solution of (31) for specific $\mathbf{\Delta}^{+-}_{\mathcal{K}_{F_1}}$, it is only needed to evaluate norms at $2^{|\mathcal{K}_{F_1}| \cdot N - 1}$ points (i.e., 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 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 (31) the bound can be over-approximated as

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

where $\lambda^{max}_{\tilde{\mathbf{O}}^\dagger_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}}$ denotes the maximal singular value of 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 matrix $\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$ (and this is non-zero as it is a full rank matrix).

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

However, to obtain the bound, in the general case the number of times that equation (31) needs 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 $\mathbf{A}\times\mathbf{C}\in\mathbb{R}^{n\times n}\times\mathbb{R}^{p\times n}$ (i.e., 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 $q_{max}=\lceil p/2-1\rceil$ [19]; in this case $s=p$, and thus F can only take the value 0, meaning that the error needs to be evaluated in $p\cdot 2^{n-1}$ if p is an odd number, or $\frac{p(p-1)}{2}2^{2n-1}$ if the system has an even number of sensors. This effectively limits the above described exhaustive search for systems with large number of states or sensors. In this case it is possible to utilize a more conservative bound introduced in [35], which significantly reduces the complexity of the procedure used for the computation.

D. Evaluation

To evaluate conservativeness of the error bound described in the previous subsection, two types of systems are considered – systems with $n=10$ states and $p=5$ sensors, and with $n=20$ states and $p=11$ sensors. For each system type, 100 systems were generated with measurement models satisfying that the rows of the $\mathbf{C}$ matrix have unit magnitude and matrices $\mathbf{\Delta}$ had elements between 0 and 2. 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. Attacks and noise profiles were chosen randomly assuming uniform distribution of the following: (a) The number of attacked sensors between 0 and 2 for systems with 5 sensors, and between 0 and 5 for systems with 11 sensors, (b) Attack vectors on the compromised sensors between -10 and 10, chosen independently for each attacked sensor, and (c) Noise realizations between the noise bounds specified by matrices $\mathbf{\Delta}$.

The considered case was with the window size N equal to the number of system states (i.e., $N=n$). Comparison between the bounds computed as described in the previous section and simulation results are shown in Fig. 2 and Fig. 3. Fig. 2(a), Fig. 2(b) and Fig. 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 – i.e., $\max_{i=1:1000}\|\Delta\mathbf{x}_{\mathfrak{S}}\|_2$, and the computed error bound $MAX\_\|\Delta\mathbf{x}_{\mathfrak{S}}\|_2$ for the system. Thus, the relative estimation error is considered, defined for each system $\mathfrak{S}$ as

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

A histogram of the relative errors for both types of systems are presented in Fig. 2(c) and Fig. 3(b). For the systems with $n=10$ states the maximal relative error reaches almost 20% of computed bounds, while for larger system (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 especially illustrated in histograms of relative estimation errors for systems with different size. As in the histograms from Fig. 2(c) and Fig. 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 smaller number of states (e.g., $n = 1, 2, 3$) we were able to generate initial states and attack vectors for which the computed bounds are tight – i.e., 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 non-compromised sensors; the attacker's goal was to maximize the state-estimation error when the proposed attack-resilient state estimation error is used.

II. 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 [36], a fully electric Unmanned Ground Vehicle (UGV) shown in Fig. 4(a). In a tethered mode, the robot can be fully tele-operated from the Operator Control Unit (OCU). However, in our scenario the operator only specifies the desired vehicle speed, while the on-board 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 (Fig. 4(b)) [37]. Here, F_l and F_r denote forces on the left and right set of wheels respectfully, 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 in order to make a turn it is necessary to generate enough torque to overcome the sticking force S_l. Therefore, when $\frac{B}{2}|F_l - F_r| \geq S_l$ the wheels start to slide sideways (i.e., the vehicle begins to turn). Consequently, if it is assumed that the wheels do not slip, the dynamical model of the vehicle can be specified as

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

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

Finally, to estimate the state of the vehicle for cruise control (i.e., its speed and position), three sensors are employed – two speed encoders, one on each sets of wheel side, and a GPS. The GPS provides time-stamped global position and speed, while from the encoders the rotation angle can be obtained (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 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 model 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 6-state linear model of this low-level electromechanical system based on the model from [37] was derived, which is then used to obtain a local state (i.e., 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 Fig. 4(c)). Since the speed of the vehicle is bounded, the attack-resilient state-estimator from (14) can be formulated as a mixed linear integer programming (MILP) problem

$$\begin{aligned} \min_{\gamma,\mathbf{E},\mathbf{x}} \quad & \mathbf{1}_p^\top \gamma \\ -\delta_k \preceq \mathbf{y}_k - \mathbf{C}\mathbf{A}^k\mathbf{x} - \mathbf{e}_k \preceq \delta_k, & \quad k = 0, ..., N-1, \\ -\gamma_j \alpha \cdot \mathbf{1}_N' \preceq \mathbf{E}_j' \preceq \gamma_j \alpha \cdot \mathbf{1}_N', & \quad j = 1, ..., p, \end{aligned}$$

where $\mathbf{E}_j'$ and $\mathbf{e}_k$ denote the j^{th} row and k^{th} column of the matrix $\mathbf{E} \in \mathbb{R}^{p\times N}$, respectively. Here, $\gamma = [\gamma_1, \dots, \gamma_p] \in \{0,1\}^p$ are binary optimization variables representing, for each sensor j, whether the sensor is considered *attacked* ($\gamma_j = 1$) 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 mph, 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 can not be larger than the maximal speed.

The developed resilient controller is executed on top of Linux OS and the Robot Operating System (ROS) middleware [38]. ROS is a meta-operating system that facilitates development of robotic applications using a publish/subscribe mechanism in which a master superintends every operation. Associated with each sensor there 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 measurements (called topics) through the master, and sends inputs to the motor driver to maintain the desired cruise speed. The tool ROSLab [39] was used to describe the architecture of the control system.

Experiments: Fig. 5 presents a deployment of the robot during experiments run on a tiled uneven surface and an uneven grass field. From the developed GUI, 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 Fig. 6. Fig. 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 encoder left.

However, as shown in Fig. 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 Fig. 6), even when one of the sensors is under attack the robot cannot reach the desired state (e.g., it can even be forced to stop). Videos of the LandShark experiments can be seen at [40].

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 (i.e., 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 $100\ \mu s$ (as shown in Fig. 7), while 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 measuring error of 0.5%. Thus, since the maximal achievable vehicle speed is $20\ m/s$, it follows that for both encoders $\delta_{k,2} = \delta_{k,2} \leq 0.1\ m/s$. 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.

A. Assurance Case for the Resilient Cruise Control Implementation

In a complex CPS design project, when a large team is engaged in design and V&V (i.e., 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 means to organize the evidence into a coherent argument that captures what evidence is available, what assumptions have been made in the design process, how each piece of evidence contributes to the overall assurance, etc. 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 [41].

In a straightforward generalization from [42], 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 PID controller. This part of the assurance case can be referred to as the control-level argument, since 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 our 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 less 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. Finally, *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 sub-claims. The first one covers the platform-independent implementation of the RSE algorithm and PID controller, implemented as a *step function* periodically invoked by the platform. The second sub-claim considers the deployment of the step function within a platform-specific wrapper, which handles periodic invocation of the step function, its connection to the streams of sensor data, and makes speed estimates available to other modules in the system. Arguments for both sub-claims are instances of the model-manipulation strategy. The step function is obtained using Simulink Coder, and which has been verified using the methods introduced in [43], [44]. 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 [45] and supplies a proof that (a) the wrapper subscribes to the sensor topics as specified in the architectural model, and that subscribed values are passed to the parameters of the step function, and also that (b) the step function is invoked with the period specified in the architectural model. This proof is used as evidence for the technique sub-claim, and review of the architectural model is performed as evidence for the model sub-claim.

III. Discussion and Future Work

In this article, methods to provide performance guarantees in CPS 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 into 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 (HACMS) program. Some of the platform assumptions made in the argument have been claims delivered by other parts of the overall assurance case.

Note that during the control design phase for resilient CPS, the designers are usually facing limitations of the platform, as 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 additional sensor payload to the platform, such as odometers, IMUs, and GPS in the LandShark case study. This, on the other hand would assume that 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, 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.

Furthermore, 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. Thus, an avenue for future work would be to provide 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.

A. Acknowledgments

This submission is part of the High Assurance Cyber Military Systems special issue. 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 [26], [41], [35].

References

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

[2] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *Security & Privacy, IEEE*, 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, "German Steel Mill Cyber Attack," *Industrial Control Systems*, Dec 2014. [Online]. Available: https://ics.sans.org/media/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf

[5] K. Zetter, "A Cyberattack Has Caused Confirmed Physical Damage for the Second Time Ever," 2015. [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 *2010 IEEE Symposium on Security and Privacy (SP)*, 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 *Proceedings of USENIX Security*, 2011.

[8] A. Greenberg, "Hackers Remotely Kill a Jeep on the Highway," 2015. [Online]. Available: http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[9] G. Jaffe and T. Erdbrink, "Iran says it downed U.S. stealth drone; Pentagon acknowledges aircraft downing," *The Washington Post*, December 2011.

[10] S. Peterson and P. Faramarzi, "Iran hijacked US drone, says Iranian engineer," *Christian Science Monitor, December*, vol. 15, 2011.

[11] D. Shepard, J. Bhatti, and T. Humphreys, "Drone hack," *GPS World*, vol. 23, no. 8, pp. 30–33, 2012.

[12] J. S. Warner and R. G. Johnston, "A simple demonstration that the global positioning system (GPS) is vulnerable to spoofing," *Journal of Security Administration*, 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 *Proceedings of the 18th ACM conference on Computer and Communications Security*, ser. CCS '11, 2011, pp. 75–86.

[14] D. Shepard, J. Bhatti, and T. Humphreys, "Drone hack: Spoofing attack demonstration on a civilian unmanned aerial vehicle," *GPS World*, 2012.

[15] 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*. Springer, 2013, pp. 55–72.

[16] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems," in *Proceedings of the 1st international conference on High Confidence Networked Systems*, ser. HiCoNS '12, 2012, pp. 55–64.

[17] R. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," *Proceedings of the IFAC World Congress*, pp. 90–95, 2011.

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

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

[20] S. Sundaram, M. Pajic, C. Hadjicostis, R. Mangharam, and G. Pappas, "The Wireless Control Network: Monitoring for malicious behavior," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 5979–5984.

[21] F. Miao, M. Pajic, and G. Pappas, "Stochastic game approach for replay attack detection," in *IEEE 52nd Annual Conference on Decision and Control (CDC)*, 2013, pp. 1854–1859.

[22] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on scada systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1396–1407, 2014.

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

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

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

[26] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. Pappas, "Robustness of attack-resilient state estimators," in *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2014, pp. 163–174.

[27] 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 *Proceedings of the 2015 American Control Conference*, 2015.

[28] Y. Shoukry, P. Nuzzo, N. Bezzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, "A satisfiability modulo theory approach to secure state reconstruction in differentially flat systems under sensor attacks," *arXiv preprint arXiv:1509.03262*, 2015.

[29] P. Antsaklis and A. Michel, *Linear Systems*. McGraw Hill, 1997.

[30] Y. Shoukry and P. Tabuada, "Event-triggered state observers for sparse sensor noise/attacks," *arXiv preprint arXiv:1309.3511*, 2013.

[31] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE, Special Issue on Technology of Networked Control Systems*, vol. 95, no. 1, pp. 138 – 162, 2007.

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

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

[34] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[35] M. Pajic, P. Tabuada, I. Lee, and G. Pappas, "Attack-resilient state estimation in the presence of noise," in *54th IEEE Annual Conference on Decision and Control (CDC)*, Dec 2015, pp. 5827–5832.

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

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

[38] 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 *Proceedings of the Open-Source Software workshop at the International Conference on Robotics and Automation (ICRA)*, 2009.

[39] N. Bezzo, J. Park, A. King, P. Gebhard, R. Ivanov, and I. Lee, "Demo abstract: Roslab – a modular programming environment for robotic applications," in *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2014, p. 214.

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

[41] J. Weimer, O. Sokolsky, N. Bezzo, and I. Lee, "Towards assurance cases for resilient control systems," in *IEEE International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, 2014, pp. 1–6.

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

[43] M. Pajic, J. Park, I. Lee, G. J. Pappas, and O. Sokolsky, "Automatic verification of linear controller software," in *Proceedings of the 12th International Conference on Embedded Software*, ser. EMSOFT '15, 2015, pp. 217–226.

[44] 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)*. Springer, 2016, pp. 662–679.

[45] The Coq development team, *The Coq proof assistant reference manual*, LogiCal Project, 2004, version 8.0. [Online]. Available: http://coq.inria.fr

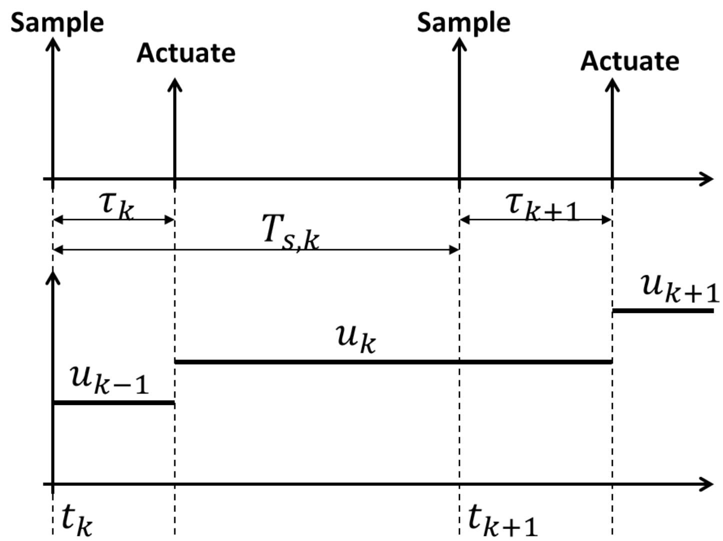
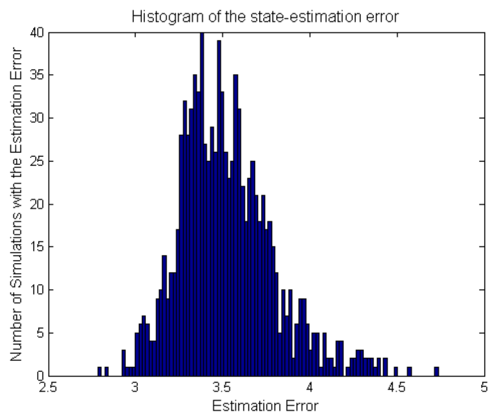
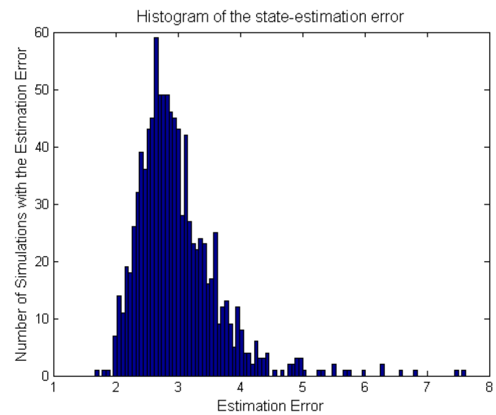


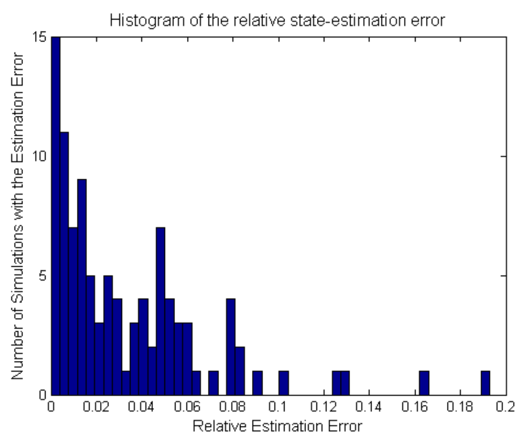
Fig. 1. Scheduling sampling and actuation.



(a) Histogram for a system with the obtained error bound equal to 41.43

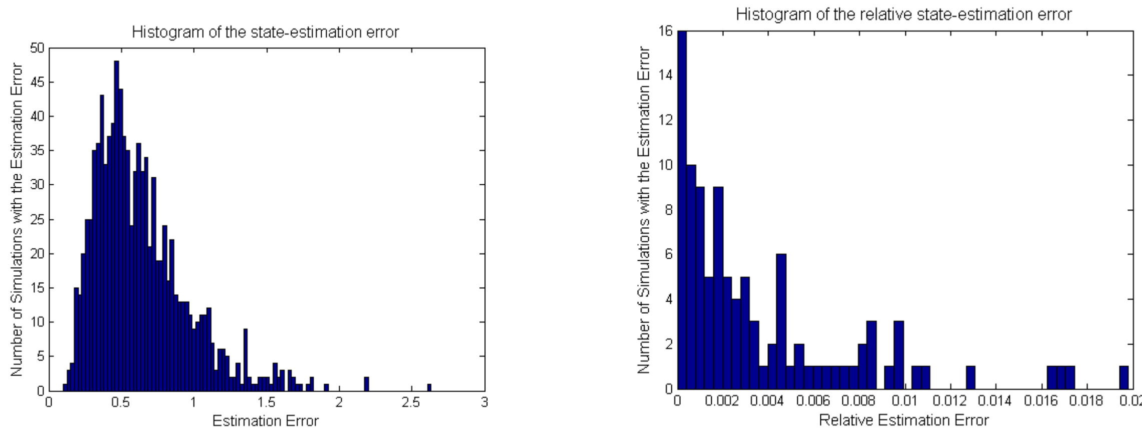


(b) Histogram for a system with the obtained error bound equal to 35.74



(c) Histogram of the maximal relative state-estimation error for all 100 system

Fig. 2. Simulation results for 1000 runs of 100 randomly selected systems with $n = 10$ states and $p = 5$ sensors.



(a) Histogram for a system with the obtained error bound equal to 155.98

(b) Histogram of the maximal relative state-estimation error for all 100 system

Fig. 3. Simulation results for 1000 runs of 100 randomly selected systems with $n = 20$ states and $p = 11$ sensors.

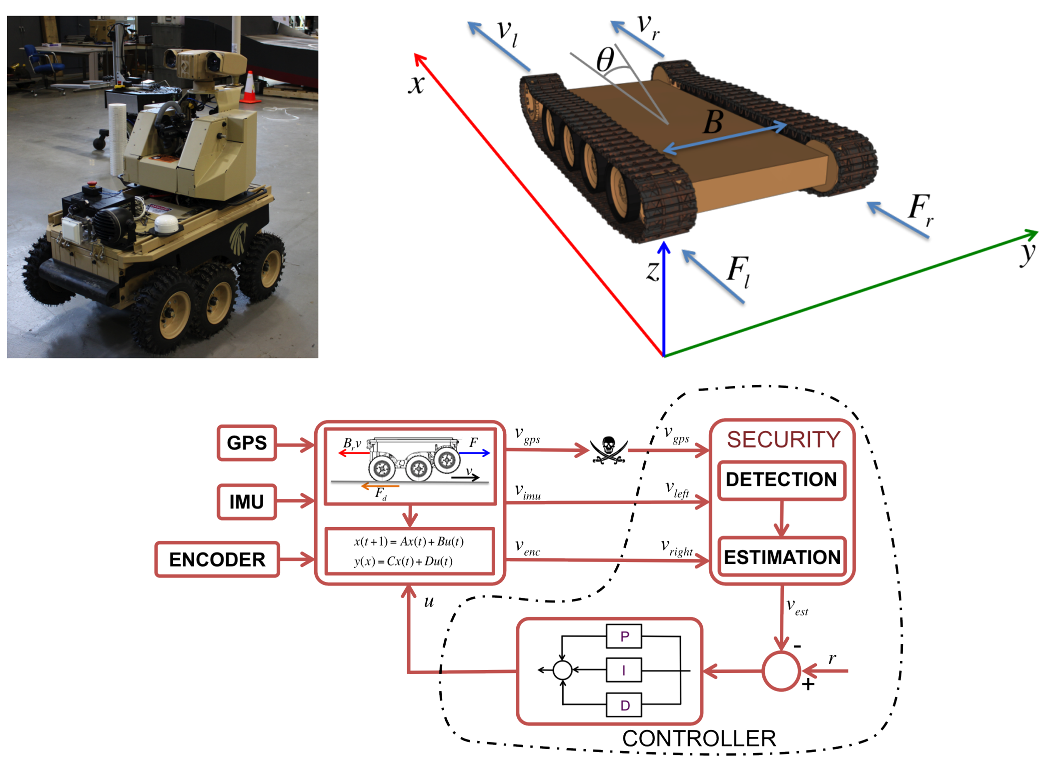


Fig. 4. LandShark unmanned ground vehicle; (a) The vehicle; (b) Coordinate system and variables used to derive the model; (c) Control system diagram used for cruise control.

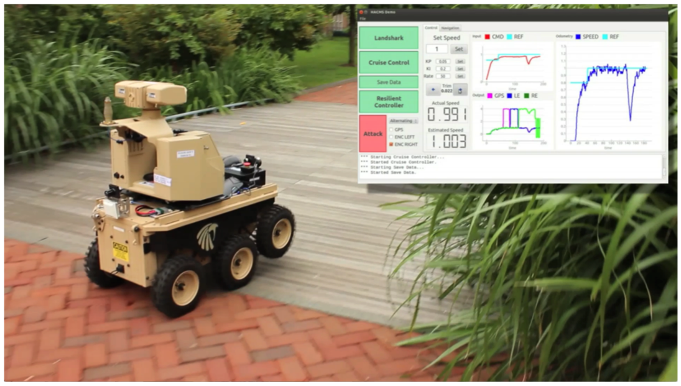


Fig. 5. Deployment of the LandShark on a tiled pathway. The picture in the picture displays the user interface used in experiments.

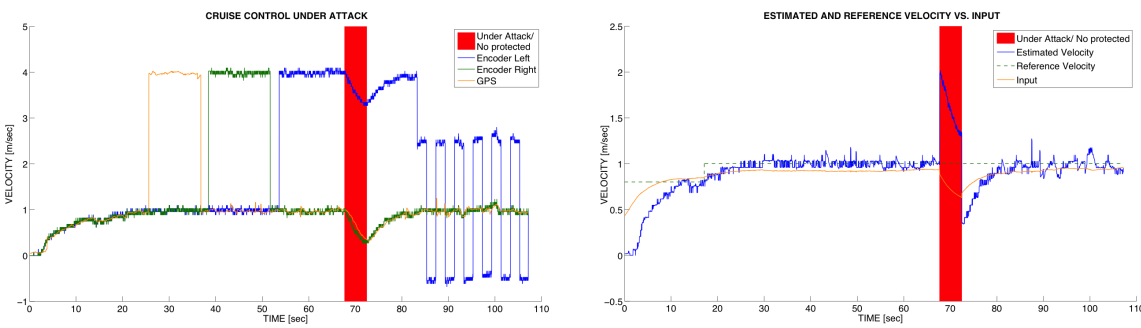


Fig. 6. Experimental results; (a) Comparison of velocity estimated from the encoders' and GPS measurements; (b) Reference speed, the estimated speed, and the input applied to the motors.

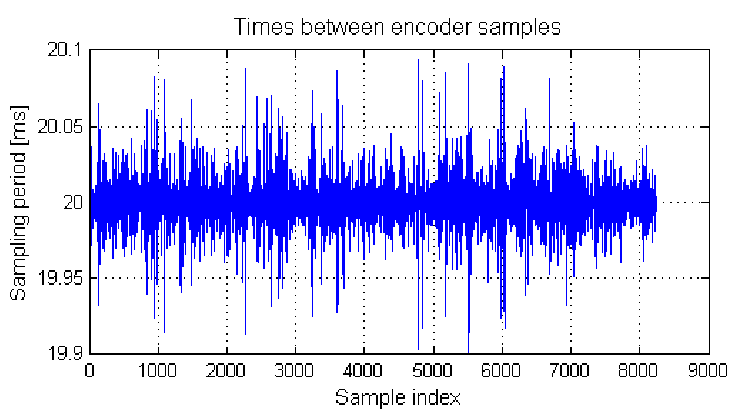


Fig. 7. Times between consecutive left encoder measurements.

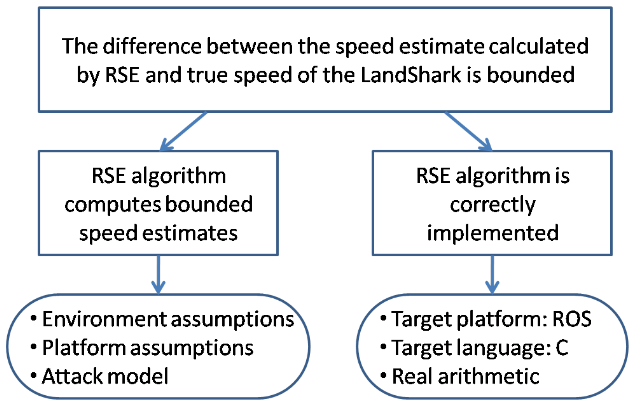


Fig. 8. Top level claims of the assurance case.

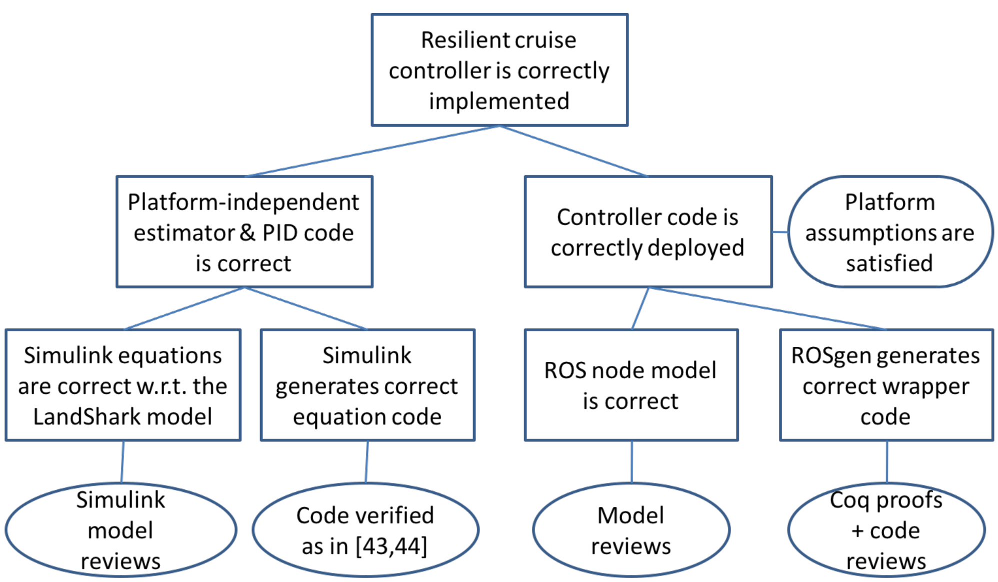


Fig. 9. Argument for the code-level claims.