Robustness of Attack-resilient State Estimators*

Miroslav Pajic
Dept. of Electrical & Systems Eng.
University of Pennsylvania
pajic@seas.upenn.edu

James Weimer Nicola Bezzo
Dept. of Computer & Information Sc.
University of Pennsylvania
{weimerj, nicbezzo}@seas.upenn.edu

Paulo Tabuada
Dept. of Electrical Eng.
UCLA
tabuada@ee.ucla.edu

Oleg Sokolsky Insup Lee
Dept. of Computer & Information Sc.
University of Pennsylvania
{sokolsky, lee}@cis.upenn.edu

George J. Pappas
Dept. of Electrical & Systems Eng.
University of Pennsylvania
pappasg@seas.upenn.edu

ABSTRACT

The interaction between information technology and physical world makes Cyber-Physical Systems (CPS) vulnerable to malicious attacks beyond the standard cyber attacks. This has motivated the need for attack-resilient state estimation. Yet, the existing state-estimators are based on the non-realistic assumption that the exact system model is known. Consequently, in this work we present a method for state estimation in presence of attacks, for systems with noise and modeling errors. When the the estimated states are used by a state-based feedback controller, we show that the attacker cannot destabilize the system by exploiting the difference between the model used for the state estimation and the *real* physical dynamics of the system. Furthermore, we describe how implementation issues such as jitter, latency and synchronization errors can be mapped into parameters of the state estimation procedure that describe modeling errors, and provide a bound on the state-estimation error caused by modeling errors. This enables mapping control performance requirements into real-time (i.e., timing related) specifications imposed on the underlying platform. Finally, we illustrate and experimentally evaluate this approach on an unmanned ground vehicle case-study.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Unauthorized access*; C.3 [**Special-purpose and Application-based Systems**]: Process control systems

1. INTRODUCTION

Tight coupling of computation and communication substrates with sensing and actuation components in Cyber-Physical Systems (CPS) has introduced significant changes in the system design process; the heterogeneity of these systems has challenged the standard design methods that completely ignore cross-cutting constraints, as component-level understanding usually does not translate to the system level. A great example for CPS are modern vehicles that present a complex interaction of a large number of embedded Electronic Control Units, interacting with each other over different types of networks. In addition, there is a current shift in vehicle architectures, from isolated control systems to more open automotive architectures that would introduce new services such as remote diagnostics and code updates, and vehicle-to-vehicle communication.

Until recently, security of CPS (and embedded control systems before) has usually been an afterthought. However, the increasing set of functionalities, network interoperability, and system design complexity may introduce security vulnerabilities that are easily exploitable. The interaction between information technology and physical world have made CPS vulnerable to malicious attacks beyond the standard cyber attacks [5]. As shown in [11, 6], using simple methods an attacker can disrupt the operation of a car to either disable the vehicle or hijack it. This problem is even more emphasized with

*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.

the rise of vehicle autonomy; thus, criticality analysis for various automotive components will have to be completely re-done. Similarly, attacks on CPS could hamper the critical infrastructure with undesired consequences as illustrated in the Maroochy Water breach [18] and StuxNet virus attack on a SCADA system used in industrial processes control [7].

Relying exclusively on cyber-security techniques for securing CPS is insufficient. This is highlighted in cases when non-invasive sensor attacks occur – i.e., when the physical environment around a sensor is compromised to allow for injection of a malicious signal [17]. For example, non-invasive attacks on Anti-lock Braking Systems are presented in [17]. In addition, attacks on GPS sensors by spoofing a GPS to misguide a yacht off route are demonstrated in [2], while [22] presents the steps and equipment required for GPS spoofing.

These results have spanned research into control-level techniques that address the problem of state estimation and intrusion detection under attacks on the environment of the controller, such as attacks on sensors, actuators and communication networks (e.g., [19, 21, 15, 9, 20, 13]). Attack-resilient state estimation has drawn considerable attention since knowing the system's state even when some components have been compromised would allow for the use of the same controllers as in the case without attacks. For deterministic linear systems this problem has been recently mapped into l_0 optimization problem [9, 15]. However, the proposed techniques consider systems without noise and for which the exact model of the system's dynamics is known. As a result, it is not clear whether most of the resilience guarantees hold when these assumptions are violated.

This problem is even more emphasized when issues inherently present in any system implementation are taken into account. For example, real-time issues such as sampling and actuation jitter, and synchronization errors between system components directly affect the controlled plant's model that should be used for state estimation. Consequently, there is a need to provide a robust method for attack-resilient state estimation in the presence of noise and modeling errors. This would also allow for 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.

We address this problem in the paper. Building on the work from [9] we present a procedure for attack-resilient state estimation in presence of noise and modeling errors. We show that the attacker cannot destabilize the system by exploiting the difference between the model used for state estimation and the *real* physical dynamics of the system. Furthermore, we describe how implementation issues such as jitter, latency and synchronization errors can be mapped into parameters of the state estimation procedure (describing modeling errors), and provide a bound on the state-estimation error due to the modeling error. This effectively enables mapping control performance requirements into real-time (i.e., timing related) specifications imposed on the underlying platform. Finally, we illustrate the use of this approach on an autonomous robot case-study.

The rest of the paper is organized as follows. In Section 2, we present an l_0-based state-estimator for deterministic linear systems, before describing common modeling errors caused by system implementation. Section 3 presents a resilient state estimator that can be used in systems with modeling errors. In Section 4, we show that the maximal state-estimation error is bounded, before we present a procedure that can be used to compute a bound. We evaluate this procedure in Section 5 and illustrate its use on unmanned ground vehicle (UGV) case study (Section 6). Finally, we provide concluding remarks in Section 7.

1.1 Notation and Terminology

For a set $\mathcal{S}$, $|\mathcal{S}|$ denotes the cardinality (i.e., size) of the set, while for two sets $\mathcal{S}$ and $\mathcal{R}$, we use $\mathcal{S} \setminus \mathcal{R}$ to denote the set of elements in $\mathcal{S}$ that are not in $\mathcal{R}$. In addition, for a set $\mathcal{K} \subset \mathcal{S}$, we specify the complement set of $\mathcal{K}$ with respect to $\mathcal{S}$ as $\mathcal{K}^{\complement}$ – i.e., $\mathcal{K}^{\complement} = \mathcal{S} \setminus \mathcal{K}$. We use $\mathbb{R}$ to denote the set of reals, and $\mathbf{1}'_N$ to denote the row vector of size N containing all ones.

We use $\mathbf{A}^T$ to indicate the transpose of matrix $\mathbf{A}$, while the i^{th} element of a vector $\mathbf{x}_k$ is denoted by $\mathbf{x}_{k,j}$. For vector $\mathbf{x}$ and matrix $\mathbf{A}$, we use $|\mathbf{x}|$ and $|\mathbf{A}|$ to denote the vector and matrix whose elements are absolute values of the initial vector and matrix, respectively. Also, for matrices $\mathbf{P}$ and $\mathbf{Q}$, $\mathbf{P} \preceq \mathbf{Q}$ specifies that the matrix $\mathbf{P}$ is *element-wise* smaller than or equal to the matrix $\mathbf{Q}$.

For a vector $\mathbf{e} \in \mathbb{R}^p$, *the support* of the vector is the set

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

while the l_0 norm of vector $\mathbf{e}$ is the size of supp($\mathbf{e}$) – i.e., $\|\mathbf{e}\|_{l_0} = |\mathrm{supp}(\mathbf{e})|$. For a matrix $\mathbf{E} \in \mathbb{R}^{p \times N}$, we use $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N$ to denote its columns and $\mathbf{E}'_1, \mathbf{E}'_2, ..., \mathbf{E}'_p$ to denote its rows. We define the *row support* of matrix $\mathbf{E}$ as the set

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

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

2. MOTIVATION AND PROBLEM DESCRIPTION

Consider a Linear-Time Invariant (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 $\mathcal{S} = \{s_1, s_2, ..., s_p\}$. 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,[1] while $\mathbf{e} \in \mathbb{R}^p$ denotes the attack vector. To model attacks on sensors, we assume that sensors with indices in set $\mathcal{K} \subseteq \{1, 2, ..., p\}$ are under attack. 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 thus $\text{supp}(\mathbf{e}_k) \subseteq \mathcal{K}$, $\forall k \geq 0$.

Remark 1. *The setup presented in this paper can be easily extended to include attacks on system actuators. In this case additional vector e_k^a is added to the plant input at each step $k \geq 0$. As shown in [8], 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 sensors and actuators are compromised. Consequently, the analysis and results presented in this paper can be easily extended to the case when a subset of the actuators is also under attack.*

2.1 Attack-resilient State Estimation for Noiseless 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 [9]. 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 is computed as the minimization argument of the following optimization problem[2]

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

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

$$\tilde{\mathbf{y}}_k = \mathbf{y}_k, \qquad k = t-N+1$$

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

In addition, $\Phi_N : \mathbb{R}^n \to \mathbb{R}^{p \times N}$ is a linear mapping defined as $\Phi_N(\mathbf{x}) = \Big[\mathbf{Cx}|\mathbf{CAx}|\dots|\mathbf{CA}^{N-1}\mathbf{x}\Big]$, 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.,

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

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 $\text{rowsupp}(\mathbf{E}_{t,N}) \subseteq \mathcal{K}$, for all N and t such that $N \geq 0$ and $t \geq N-1$.

As shown in [8, 9], 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 $\mathbf{x} \in \mathbb{R}\backslash\{\mathbf{0}\}$,

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

We use q_{max} to denote the maximal number of compromised sensors for which the system's state can be recovered after N steps despite attacks on sensors.[3] 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. Note that for these systems q_{max} does not decrease with N, and due to Cayley-Hamilton theorem [3] 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 $\mathbf{C}$).

2.2 Sources of Modeling Errors

Besides measurement and process noise, vectors $\mathbf{v}_k$ and $\mathbf{w}_k$ in (1) can be used to capture any deviation in the plant model (1) from the real dynamics of the controlled physical system. Here, we present some of the common modeling errors introduced by non-idealities of control system implementation and limitations of the utilized computation and communication platforms. Specifically, we focus on the modeling errors caused by sampling and computation/actuation jitter, and synchronization errors between system components in scenarios where continuous-time plants are being controlled.

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

[1] Later we will assume that the noise vectors are constrained in certain ways. Furthermore, we will use $\mathbf{v}$ and $\mathbf{w}$ to capture different types of modeling errors which may be caused by some implementation (e.g., real-time) issues. We will address this in more details in Section 2.2.

[2] The state $\mathbf{x}_t$ can then be obtained from $\mathbf{x}_{t-N+1}$ and 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.

[3] The size of the utilized measurement history N is considered to be an input parameter to the resilient-state estimator. In the general case we should use the notation $q_{max,N}$.

this, consider a standard continuous-time plant model

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}_c\mathbf{x}(t),\end{aligned} \tag{4}$$

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.

We 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. We denote the k^{th} sampling period of the plant by $T_{s,k} = t_{k+1} - t_k$, and note that the the input signal will have the form shown in Fig. 1(b). Using the approach from [10, 23], we describe the system as

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{u}(t),\\ \mathbf{y}(t) &= \mathbf{C}_c\mathbf{x}(t), \quad t \in [t_k + \tau_k, t_{k+1} + \tau_{k+1}),\\ \mathbf{u}(t^+) &= \mathbf{u}_k, \quad t \in \{t_k + \tau_k, k = 0, 1, 2, \ldots\}\end{aligned} \tag{5}$$

where $\mathbf{u}(t^+)$ is a piecewise continuous function that only changes values at time instances $t_k + \tau_k$, $k \geq 0$. From the above equation, the discretized model of the system can be represented as [3]

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{B}_k^-\mathbf{u}_{k-1}\\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k,\end{aligned} \tag{6}$$

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

$$\begin{aligned}&\mathbf{A}_k = e^{\mathbf{A}_c T_{s,k}},\\ &\mathbf{B}_k = \int_0^{T_{s,k}-\tau_k} e^{\mathbf{A}_c\theta}\mathbf{B}_c d\theta, \ \ \mathbf{B}_k^- = \int_{T_{s,k}-\tau_k}^{T_{s,k}} e^{\mathbf{A}_c\theta}\mathbf{B}_c d\theta.\end{aligned} \tag{7}$$

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 $\tau_k = 0$

$$\mathbf{A} = e^{\mathbf{A}_c T_s}, \ \ \mathbf{B} = \int_0^{T_s} e^{\mathbf{A}_c\theta}\mathbf{B}_c d\theta, \tag{8}$$

Hence, by comparing the discrete-time models (1) and (6), 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

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

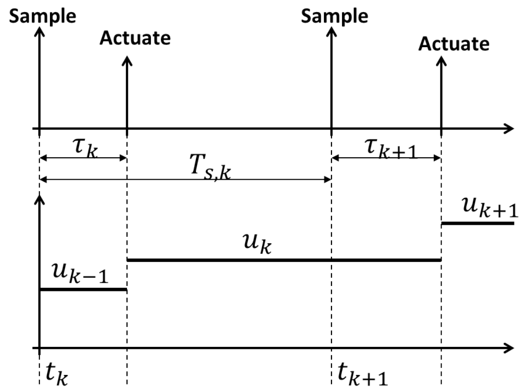


Figure 1: Scheduling sampling and actuation.

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.

2.2.1 Effects of Synchronization Errors

Consider a setup where the sensors do not have a common clock source, resulting in a bounded error between the individual clocks at sensors.[4] In this case, although scheduled to measure corresponding plant outputs at the same time-instance t_k, each sensor s_j will actually perform the measurement at time $t_{k,j}$. Hence, 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 $\mathbf{C}$, and the synchronization error introduces a measurement error defined as

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

Here, $\Delta t_{k,j} = t_{k,j} - t_k$ captures the synchronization error for each sensor s_j. Hence, 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 due to synchronization errors between sensors.

2.3 Problem Description

The existence of modeling errors[5] 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

[4] To simplify the presentation we only describe the case where there possibly exist synchronization errors between sensors, because the same approach can be extended to scenarios where there exist synchronization errors between plant actuators.

[5] In the rest of this paper, unless otherwise specified we will include process and measurement noise as part of modeling errors. We will also use the term *noise* to capture the modeling errors – i.e., discrepancy between the model used to design the state-estimator and real dynamics of the controlled system.

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, in this paper we focus on the following problems

- How can we design attack-resilient state estimators in presence of noise and modeling errors?
- Can the attacker exploit the modeling errors in the state-estimator's design in order to destabilize the system when the state estimates are used for control by a *stable* state-based feedback controller?
- Is it possible to obtain a bound on the worst-case performance degradation of the introduced resilient-state estimator due to bounded modeling errors?

3. RESILIENT STATE ESTIMATION IN PRESENCE OF MODELING ERRORS

As illustrated in Section 2, 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 paper (unless otherwise stated) we will assume that in (1) $\mathbf{u}_k = \mathbf{0}$ for all $k \geq 0$. In addition, to further simplify the notation we consider the case for $t = N-1$, meaning that our goal is to obtain $\mathbf{x}_0$, and we denote the matrices $\mathbf{Y}_{t,N}, \mathbf{E}_{t,N}$ and $\mathbf{\Phi}_N(\mathbf{x})$ as $\mathbf{Y}, \mathbf{E}$ and $\mathbf{\Phi}(\mathbf{x})$, respectively.

We assume that the state of the plant at $k = 0$ is $\mathbf{x}_0$ and that the system evolves 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 on the sensors from set $\mathcal{K} = \{s_{i_1}, ..., s_{i_q}\} \subseteq \mathcal{S}$, where $|\mathcal{K}| \leq q_{max}$, and the corresponding matrix $\mathbf{E} = [\mathbf{e}_0|\mathbf{e}_1|\dots|\mathbf{e}_{N-1}]$. Furthermore, we assume that $|\mathbf{w}_k| \preceq \epsilon_{w_k}$ and $|\mathbf{v}_k| \preceq \epsilon_{v_k}$ for $k = 0, 1, ..., N-1$, and we 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 with noise. Finally, we use $\bar{\mathbf{Y}} = [\bar{\mathbf{y}}_0|\bar{\mathbf{y}}_1|...|\bar{\mathbf{y}}_{N-1}]$ to denote 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 $\bar{\mathbf{y}}_k = \mathbf{C}\mathbf{A}^k\mathbf{x}_0 + \mathbf{e}_k$, $k = 0, 1, ..., N-1$).

We consider the following optimization problem

$$\begin{aligned} P_0(\mathbf{Y}): \qquad & \min_{\mathbf{E},\mathbf{x}} \|\mathbf{E}\|_{l_0} \\ s.\ t. \quad & \mathbf{E} = \mathbf{Y} - \Phi(\mathbf{x}) \end{aligned} \tag{9}$$

As described in Section 2.1,

$$(\mathbf{x}_0, \mathbf{E}) = arg\max P_0(\bar{\mathbf{Y}}) \tag{10}$$

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 $(\mathbf{x}_0, \mathbf{E})$ 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 we consider the following problem that relaxes the equality constraint from (9) by including a noise allowance

$$\begin{aligned} P_{0,\boldsymbol{\Delta}}(\mathbf{Y}): \qquad & \min_{\mathbf{E},\mathbf{x}} \|\mathbf{E}\|_{l_0} \\ s.\ t. \quad & |\mathbf{Y} - \Phi(\mathbf{x}) - \mathbf{E}| \preceq \boldsymbol{\Delta} \end{aligned} \tag{11}$$

In the above problem, the matrix $\boldsymbol{\Delta} \in \mathbb{R}^{p \times N}$ contains non-negative tolerances $\delta_{j,i}$ for each sensor s_i, $i = 1, ..., p$, in each of the N steps j – i.e.,

$$\boldsymbol{\Delta} = [\delta_0|\delta_1|\dots|\delta_{N-1}]\,.$$

We use the following notation

$$\begin{aligned} (\mathbf{x}_{0,\boldsymbol{\Delta}}, \mathbf{E}_{\boldsymbol{\Delta}}) &= arg\max P_{0,\boldsymbol{\Delta}}(\mathbf{Y}_{\mathbf{w},\mathbf{v}}) \\ q_{\boldsymbol{\Delta}} &= \|\mathbf{E}_{\boldsymbol{\Delta}}\|_{l_0} \end{aligned} \tag{12}$$

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 (11) as an attack-resilient state estimator it is necessary to ensure that $P_{0,\boldsymbol{\Delta}}(\mathbf{Y})$ has a feasible point $(\mathbf{x}, \mathbf{E})$ such that $\|\mathbf{E}\|_{l_0} \leq q_{max}$.[6] This can be guaranteed with an appropriate initialization of the matrix $\boldsymbol{\Delta}$. From (1) we have that for $k = 0, 1, ..., N-1$[7]

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

If we use $|\mathbf{A}^{k-1-i}|$ to denote the matrix whose elements are absolute values of the corresponding elements of the matrix $\mathbf{A}^{k-1-i}$, then we can specify the following bound

$$\begin{aligned} |\mathbf{y}_k - \bar{\mathbf{y}}_k| &\leq |\mathbf{C}|\sum_{i=0}^{k-1}|\mathbf{A}^{k-1-i}||\mathbf{v}_i| + |\mathbf{w}_k| \\ &\leq |\mathbf{C}|\sum_{i=0}^{k-1}|\mathbf{A}^{k-1-i}|\epsilon_{v_i} + \epsilon_{w_i} = \bar{\delta}_k. \end{aligned} \tag{13}$$

Therefore, for $\delta_k \succeq \bar{\delta}_k$ ($k = 0, ..., N-1$) we have that $(\mathbf{x}_0, \mathbf{E})$ from (10) is a feasible point for the problem $P_{0,\boldsymbol{\Delta}}(\mathbf{Y}_{\mathbf{w},\mathbf{v}})$, meaning that there exists a solution of the problem – i.e., there exists $(\mathbf{x}_{0,\boldsymbol{\Delta}}, \mathbf{E}_{\boldsymbol{\Delta}})$ from (12) such that $q_{\boldsymbol{\Delta}} = q \leq q_{max}$. This means that the solution of $P_{0,\boldsymbol{\Delta}}(\mathbf{Y}_{\mathbf{w},\mathbf{v}})$ from (11) 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 $\mathbf{E}_{\boldsymbol{\Delta}}$ is not larger than q_{max}.

[6] 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.

[7] We assume that $\sum_{i=0}^{-1} \alpha_i = 0$ for any sequence of α_is.

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

In this section, we provide robustness analysis for $P_{0,\mathbf{\Delta}}(\mathbf{Y})$ optimization problem when matrix $\mathbf{\Delta}$ satisfies conditions from the previous section. We start by showing that the attacker cannot exploit the modeling errors to destabilize the system before we present a method to bound the error caused by noise and modeling errors – i.e., we provide a bound on $\|\mathbf{x}_{0,\mathbf{\Delta}} - \mathbf{x}_0\|_2$.

Consider $(\mathbf{x}_{0,\mathbf{\Delta}}, \mathbf{E}_{\mathbf{\Delta}})$ from (12), and a matrix $\mathbf{\Sigma} \in \mathbb{R}^{p\times N}$ such that

$$\mathbf{Y} - \Phi(\mathbf{x}_{0,\mathbf{\Delta}}) - \mathbf{E}_{\mathbf{\Delta}} = \mathbf{\Sigma}. \tag{14}$$

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

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

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

$$\Delta\mathbf{x} = \mathbf{x}_{0,\mathbf{\Delta}} - \mathbf{x}_0. \tag{15}$$

If we also define $\Delta\mathbf{E} = \mathbf{E}_{\mathbf{\Delta}} - \mathbf{E}$, it holds that

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

Let us denote by $\Delta\mathbf{y}_0, ..., \Delta\mathbf{y}_{N-1}$ the columns of the matrix $\mathbf{\Omega}$ (i.e., $\mathbf{\Omega} = [\Delta\mathbf{y}_0|...|\Delta\mathbf{y}_{N-1}]$). From (13) and (14) it follows that

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

Accordingly, to provide a bound on $\|\Delta\mathbf{x}\|_2$ we consider the following problem

$$\max_{\Delta\mathbf{x}} \quad \|\Delta\mathbf{x}\|_2 \tag{16}$$

$$\|\Phi(\Delta\mathbf{x}) - \mathbf{\Omega}\|_{l_0} \leq 2q \tag{17}$$

$$\mathbf{\Omega} \preceq 2\mathbf{\Delta} \tag{18}$$

Since $q \leq q_{max}$, we can increase the feasible space by relaxing constraint (17) to

$$\|\mathbf{\Omega} - \Phi(\Delta\mathbf{x})\|_{l_0} \leq 2q_{max} \tag{19}$$

Therefore, our goal is to bound $\Delta\mathbf{x}$ for which there exists $\mathbf{\Omega} \in \mathbb{R}^{p\times N}$ that satisfies (18), and for where ***at least*** $p - 2q_{max}$ rows of the matrix $\Phi(\Delta\mathbf{x}) - \mathbf{\Omega}$ are zero-rows. Let us 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 $F_c = p - 2q_{max} - F$ rows of $\Phi(\Delta\mathbf{x})$ are equal to the rows of $\mathbf{\Omega}$, which are non-zero, and we 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 $\mathcal{K}_F \cap \mathcal{K}_{F_1} = \emptyset$.

We use the following notation – for any set $\mathcal{K} \subseteq S$ we define the matrix $\mathbf{O}_{\mathcal{K}}$ as

$$\mathbf{O}_{\mathcal{K}} = \begin{bmatrix} P_{\mathcal{K}}\mathbf{C} \\ P_{\mathcal{K}}\mathbf{CA} \\ \vdots \\ P_{\mathcal{K}}\mathbf{CA}^{N-1} \end{bmatrix}. \tag{20}$$

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 $\mathcal{K}$.[8] Since $\mathcal{K}_F \subset \mathcal{S}$ contains indices of zero-rows of $\Phi(\Delta\mathbf{x})$ we have that $\mathbf{O}_{\mathcal{K}_F}\Delta\mathbf{x} = \mathbf{0}$. In addition, $\mathbf{O}_{\mathcal{K}_{F_1}}\Delta\mathbf{x} = \mathbf{\Omega}_{\mathcal{K}_{F_1}}$, 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 we define

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

Consequently, for $\Delta\mathbf{x}$ to satisfy constraints (19) and (18) there have to exist sets $\mathcal{K}_F, \mathcal{K}_{F_1} \subset \mathcal{S}$ such that

$$|\mathcal{K}_F| = F,\ |\mathcal{K}_{F_1}| = p - 2q_{max} - F, \tag{21}$$

$$\mathcal{K}_F \cap \mathcal{K}_{F_1} = \emptyset \tag{22}$$

$$\mathbf{O}_{\mathcal{K}_F}\Delta\mathbf{x} = \mathbf{0} \tag{23}$$

$$|\mathbf{O}_{\mathcal{K}_{F_1}}\Delta\mathbf{x}| \preceq 2\mathbf{\Delta}_{\mathcal{K}_{F_1}} \tag{24}$$

We consider the polyhedron $\mathbb{P}$ defined with constraints (21)-(24). Note that the point $\Delta\mathbf{x} = \mathbf{0}$ belongs to the polyhedron. In the rest of this section we will first prove that the polyhedron is bounded, before we present a procedure that can be used to provide a bound on $\max\|\Delta\mathbf{x}\|_2$. We start by introducing the following lemma.

Lemma 1. *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 $\mathcal{K}_F \cap \mathcal{K}_{F_1} = \emptyset$,*

$$rank(\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = n. \tag{25}$$

Proof. From [9], $q_{max} = \lceil s/2 - 1 \rceil$ 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 $\mathbf{O}_{\mathcal{K}_1^{\complement}} \in \mathbb{R}^{N|\mathcal{K}_1^{\complement}|\times n}$,

$$rank(\mathbf{O}_{\mathcal{K}_1^{\complement}}) = n.$$

[8] Formally, $P_{\mathcal{K}} = \begin{bmatrix} \mathbf{i}'_{k_1} \\ \vdots \\ \mathbf{i}'_{k_{|\mathcal{K}|}} \end{bmatrix}$, where $\mathcal{K} = \{s_{k_1}, ..., s_{k_{|\mathcal{K}|}}\}$ and $k_1 < k_2 < ... < k_{|\mathcal{K}|}$, and $\mathbf{i}'_j$ denotes the row vector (of appropriate size) with a 1 in its j^{th} position.

The above relation holds for any $\mathcal{K}_1^{\complement}$ with at least $p - 2q_{max}$ sensors, and hence (25) 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* (21)-(24) *is bounded.*

PROOF. We start by assuming 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 $\epsilon > 0$, $\Delta\mathbf{x} + \epsilon\mathbf{d} \in \mathbb{P}$ [4]. 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,

$$|\mathbf{O}_{\mathcal{K}_{F_1}}(\Delta\mathbf{x} + \epsilon\mathbf{d})| \preceq 2\mathbf{\Delta}_{\mathcal{K}_{F_1}} \tag{26}$$

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 (26) 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 1, $\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}$ has full rank (i.e., $\text{rank}(\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = n$), meaning that it does not have a 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 we have that maximal $\|\Delta\mathbf{x}\|_2$ is ***bounded, and the attacker can not use modeling errors and the corresponding relaxation of the l_0 optimization problem to introduce unbounded error in the attack-resilient state estimator.*** Thus, we can formulate the corollary.

COROLLARY 1. *Consider a matrix $\mathbf{K}$ such that $\mathbf{A} + \mathbf{BK}$ is stable. If $\mathbf{x}_{0,\mathbf{\Delta}}$ from* (12) *is used as state estimate by the state-feedback controller $\mathbf{K}$ (i.e., $\mathbf{u} = \mathbf{Kx}_{0,\mathbf{\Delta}}$) then the closed-loop system will remain stable when at most q_{max} sensors have been compromised.*

4.1 Bounding the State-estimation Error

Theorem 1 also facilitates bounding the error of the resilient state estimator $P_{0,\mathbf{\Delta}}(\mathbf{Y}_{\mathbf{w},\mathbf{v}})$ by starting from the following lemma.

LEMMA 2. *Consider the optimization problem*

$$\max_{\mathbf{x} \in \mathbb{Q}} f(\mathbf{x})$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function and $\mathbb{Q}$ is a bounded polyhedron. Then, there exists an optimal solution $f(\mathbf{x}^*)$ where $\mathbf{x}^*$ is a vertex of the polyhedron $\mathbb{Q}$.

PROOF. See Appendix 8. □

From the above lemma, 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. On the other hand, the vertices of the polyhedron satisfy that

$$\underbrace{\begin{bmatrix} \mathbf{O}_{\mathcal{K}_F} \\ \mathbf{O}_{\mathcal{K}_{F_1}} \end{bmatrix}}_{\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}} \cdot \Delta\mathbf{x} = \begin{bmatrix} \mathbf{0} \\ 2\mathbf{\Delta}_{\mathcal{K}_{F_1}}^{+-} \end{bmatrix}, \tag{27}$$

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 ($\text{rank}(\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = \text{rank}(\mathbf{O}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}) = n$), vertex points can be found as

$$\Delta\mathbf{x}_{ver} = (\tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}^T \tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}})^{-1} \tilde{\mathbf{O}}_{\mathcal{K}_F \cup \mathcal{K}_{F_1}}^T \begin{bmatrix} \mathbf{0} \\ 2\mathbf{\Delta}_{\mathcal{K}_{F_1}}^{+-} \end{bmatrix}. \tag{28}$$

Consequently, for any sets $\mathcal{K}_F$ and $\mathcal{K}_{F_1}$ that satisfy (21) and (22), by checking all $2^{|\mathcal{K}_{F_1}| \cdot N}$ vertices defined by (28) we can determine the maximal $\|\Delta\mathbf{x}\|_2$ 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 (28) for specific $\mathbf{\Delta}_{\mathcal{K}_{F_1}}^{+-}$, we only need 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 (18) and (19) we have to consider all such sets $\mathcal{K}_F$ and $\mathcal{K}_{F_1}$. Therefore, it is necessary to evaluate all possible values for F. From the definition $F \geq 0$. On the other hand, from (23) $\mathbf{O}_{\mathcal{K}_F}$ has a nontrivial kernel, meaning that as in the proof of Lemma 1, $F = |\mathcal{K}_F| \leq p - s \leq p - 2q_{max} - 1$.

This allows for the formulation of Algorithm 1 and proves the following theorem.

THEOREM 2. *Algorithm 1 provides an upper bound on the state estimation error $\|\mathbf{x}_{0,\mathbf{\Delta}} - \mathbf{x}_0\|_2$.*

Finally, note that the matrix $\mathbf{\Delta}$ captures several sources of modeling errors (e.g., noise, jitter, synchronization errors). Since (28) is linear in $\mathbf{\Delta}$, the estimation error bound obtained by Algorithm 1 for $\mathbf{\Delta}$ will be less than or equal to the sum of estimation error bounds computed separately for each error component. Thus, it is possible to analyze the impact of each source of modeling errors on robustness of the state estimator.

4.1.1 Complexity of Algorithm 1

The complexity of Algorithm 1 depends on the number of plant states and sensors (n and p, respectively), considered window size (i.e., history) N, and the maximal number of attacked sensors q_{max} for which it is possible to estimate the state of the noiseless system after N steps. Hence, in the general case the number of times that equation (28) needs to be solved is

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

However, 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.,

Algorithm 1 ***Design-time*** procedure used to provide a bound on $\Delta\mathbf{x}$

```
$MAX\_\|\Delta\mathbf{x}\| = 0$
for $F = 0, 1, ..., p - s$ do
    for all sets $\mathcal{K}_F \subset \mathcal{S}$ with $F$ elements do
        for all sets $\mathcal{K}_{F_1} \subset (\mathcal{S} \setminus \mathcal{K}_F)$ with $p - 2q_{max} - F$ elements do
            for all $\mathbf{\Delta}^{+-}_{\mathcal{K}_{F_1}}$ do
                Compute $\Delta\mathbf{x}_{ver}$ using (28)
                if $\|\Delta\mathbf{x}_{ver}\|_2 > MAX\_\|\Delta\mathbf{x}\|$ then
                    $MAX\_\|\Delta\mathbf{x}\| = \|\Delta\mathbf{x}_{ver}\|_2$
                end if
            end for
        end for
    end for
end for
```

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 $q_{max} = \lceil p/2 - 1 \rceil$ [9]. Note that in this case $s = p$, and thus F can only take the value 0 in Algorithm 1. On the other hand, $q_{max} = \begin{cases} \frac{p-1}{2} & p \textit{ is odd} \\ \frac{p}{2} - 1 & p \textit{ is even} \end{cases}$ and thus

$$|\mathcal{K}_{F_1}| = \begin{cases} 1, & p \textit{ is odd} \\ 2, & p \textit{ is even} \end{cases}$$

Therefore, for almost all systems with n states and p sensors, if we use $N = n$ past measurement vectors the complexity of Algorithm 1 is $p \cdot 2^{n-1}$ if p is an odd number, or $\frac{p(p-1)}{2} 2^{2n-1}$ if the system has an even number of sensors.

5. EVALUATION

To evaluate conservativeness of the error bound obtained using Algorithm 1 we consider two types of systems – systems with $n = 10$ states and $p = 5$ sensors, and with $n = 20$ states and $p = 11$ sensors. For each system type we randomly generated 100 systems 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 we evaluated the state-estimation error $\Delta\mathbf{x} = \|\mathbf{x}_{0,\mathbf{\Delta}} - \mathbf{x}_0\|_2$ 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 $\mathbf{\Delta}$.

In both simulations and Algorithm 1 executions, we considered the case when window size N is equal to the number of system states (i.e., $N = n$). Comparison between the bounds provided by Algorithm 1 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 bound provided by Algorithm 1 is an order of magnitude larger than the average state-estimation error for each system. However, for each system $\mathfrak{S}$ we are more interested in 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 error bound $MAX\_\|\Delta\mathbf{x}_{\mathfrak{S}}\|_2$ provided by Algorithm 1 for the system. Thus, we consider relative estimation error defined for each system $\mathfrak{S}$ as

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

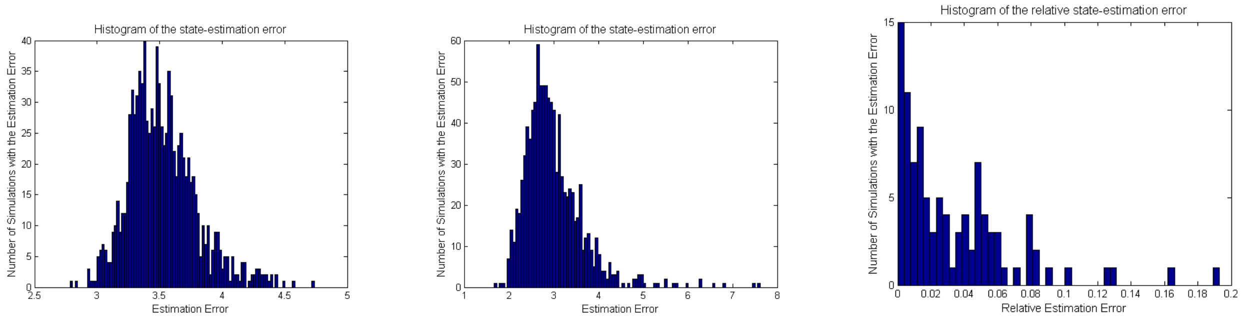
A histogram of the relative errors for both types of systems are presented in Fig. 2(c) and Fig. 3(b). For the systems with $n = 10$ states the maximal relative error reaches almost 20% of computed bounds, while for larger system (with $n = 20$ states) the maximal relative error is 2% of computed bounds.

However, it is worth noting here that conservativeness of the presented results is (at least partially) caused by the fact that for each system we only considered random initial points, 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 paper.

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), in simulations we observed a decrease in the obtained maximal relative estimation error 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 we increase the number of attack vectors, and due to the random selection of the attack vectors we reduce probabilities to incorporate a worst-case attack. On the other hand, for systems with $n = 1$ and $n = 2$ states we were able to generate initial states and attack vectors for which the bounds from Algorithm 1 are tight – i.e., the error $\|\Delta\mathbf{x}\|_2$ is equal to the obtained bounds.

6. CASE STUDY

We illustrate the development framework on a design

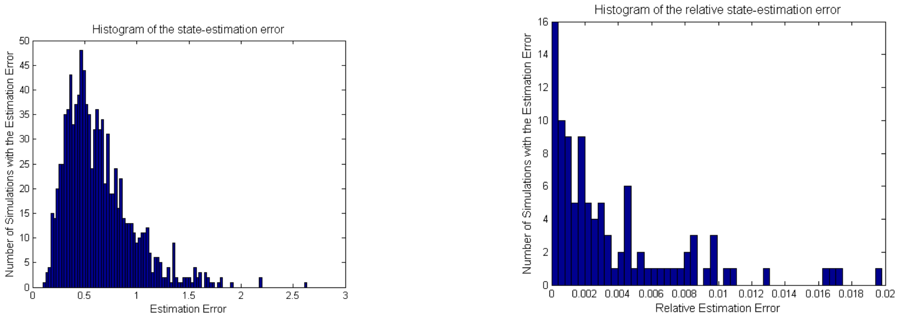


(a) Histogram for a system with error bound 41.43

(b) Histogram for a system with error bound 35.74

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

Figure 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 error bound 155.98

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

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

of secure cruise control of the LandShark vehicle [1], 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 we used the standard differential drive vehicle model (Fig. 4(b)) [12]. 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 we assume that the wheels do not slip, the dynamical model of the vehicle can be specified as

$$\dot{v} = \begin{cases} \frac{1}{m}(F_l + F_r - (B_s + B_r)v, & \text{if } \textit{turning} \\ \frac{1}{m}(F_l + F_r - B_r v, & \text{if } \textit{not turning} \end{cases}$$

$$\dot{\omega} = \begin{cases} \frac{1}{J_t}(\frac{B}{2}(F_l - F_r) - B_l \omega, & \text{if } \textit{turning} \\ 0, & \text{if } \textit{not turning} \end{cases}$$

$$\dot{\theta} = \omega$$

$$\dot{x} = v\sin(\theta), \qquad \dot{y} = v\cos(\theta)$$

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) we use three sensors – two speed encoders, one on each set of wheels, and a GPS. The GPS provides time-stamped global position and speed, while from the encoders we can obtain the rotation angle (which could 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 coming from the IMU, or use optical flow algorithms to com-

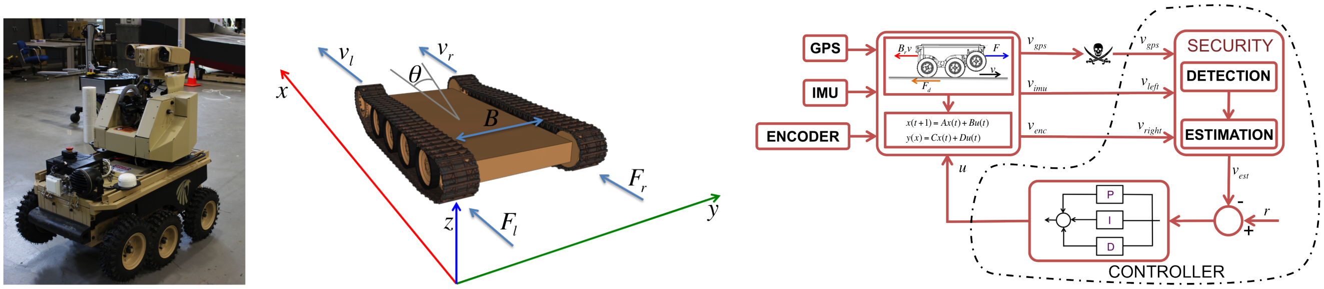


Figure 4: LandShark unmanned ground vehicle; (a) The vehicle; (b) Coordinate system and variables used to derive the model; (c) Control system diagram used for cruise control.

pute visual odometry from a camera. However, to illustrate the use (and robustness) of the attack-resilient state estimator we only used the encoders and GPS.

The above model presents a high-level model of the vehicle, describing only the motion equations. However, 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, we have also derived a 6-state linear model of this low-level electromechanical system based on the model from [12], which is then used to derive a local state (i.e., velocity) feedback controller that provides the desired F_l, F_r levels.

System Architecture

On the LandShark, the CPU that implements the state-estimation and controller procedure is connected to all sensors 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 (11) can be formulated as a mixed linear integer programming (MILP) problem

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

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.[9]

The developed resilient controller is executed on top of Linux OS and the Robot Operating System (ROS) middleware [16]. ROS is a meta-operating system that facilitates development of robotic applications using a publish/subscribe mechanism in which a master superintend every operation. There is a driver associated with each sensor, which takes care of getting time stamped informations 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 in order to maintain the desired cruise control speed.

Experiments

Fig. 5 presents a deployment of the robot during experiments run on a tiled uneven surface and a grass uneven field. From the developed GUI we demonstrate 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 has been 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 we have 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 shredded 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).[10]

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 run with a

[9] Since the robot cannot obtain speed larger than 20 mph, all sensor measurements larger than the value are obtained from compromised sensors and thus can be discarded. Hence, we can assume that elements of attack vectors can not be larger than the maximal speed.

[10] Videos of the Landshark experiments can be seen at `http://www.seas.upenn.edu/~pajic/research/CPS_security.html`.

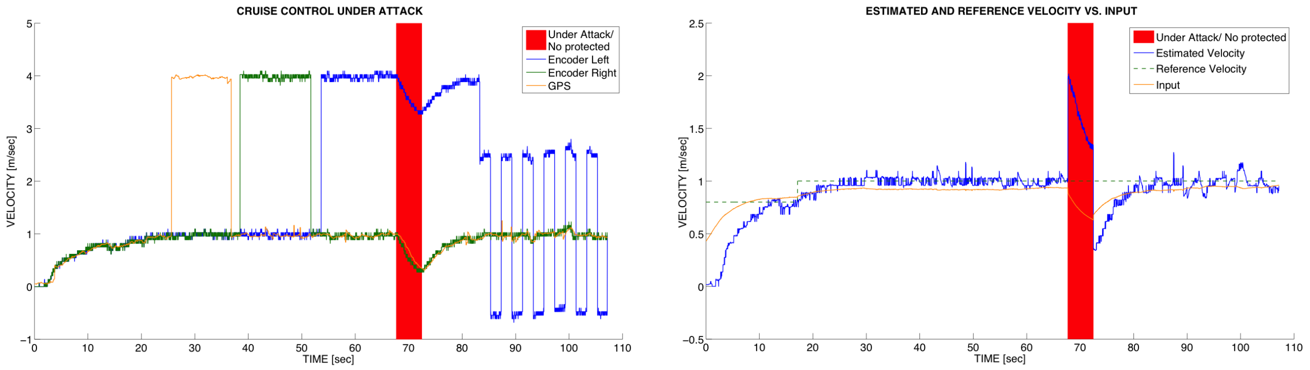


Figure 6: Experimental results; (a) Comparison of velocity estimated from the encoders' and GPS measurements; (b) Reference speed, the estimated speed, and the input applied to the motors.

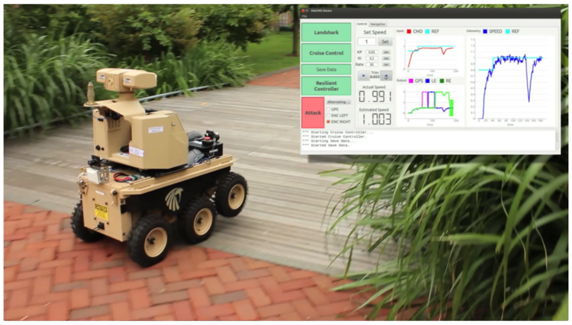


Figure 5: Deployment of the LandShark on a tiled pathway. The picture in the picture displays the user interface used in experiments.

variable period). Each sensor has its own clock and all measurements are time-stamped before being transmitted to the controller. However, 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 Fig. 7), while GPS has highly variable jitter with maximal values up to 125 ms. Therefore, it is not possible to use the idealized discrete-time model from (8), but rather the full input compensation has to be done as in (6) and (7), 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, we have that for both encoders $\delta_{k,2} = \delta_{k,2} \leq 0.1\ m/s$. For these values Alg. 1 provides a state-estimation error bound of 0.72 m/s.

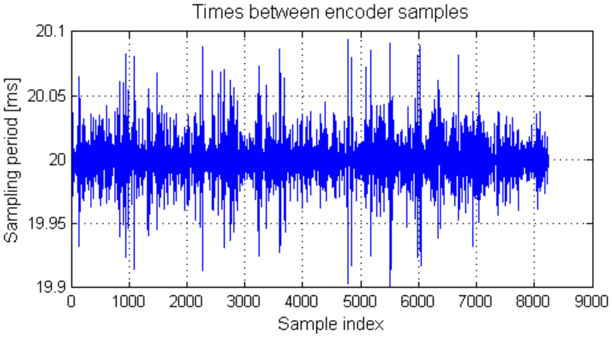


Figure 7: Times between consecutive left encoder measurements.

Note that the conservativeness of the bound was mostly caused by the large worst-case GPS sampling jitter.

7. CONCLUSION

In this paper, we have considered the problem of attack-resilient state estimation in systems with noise and where the exact model of the system dynamics is not known. We have described a l_0-norm based state estimator that can be used for these systems, and showed that the attacker cannot exploit the noise and limitations in model accuracy to destabilize the system. Furthermore, we have provided an algorithm to derive a bound for the state estimation error caused by noise and modeling errors, and presented a procedure to map these bounds into a set of implementation specifications imposed on the underlying platform. Finally, we have illustrated our approach by designing an attack-resilient constant speed cruise controller for unmanned ground vehicle.

We have shown that the presented l_0-norm optimization procedure for state estimation can be formulated as a mixed integer linear program. Although there exist efficient MILP solvers, MILPs are effectively NP hard.

Therefore, the natural next step would include transforming the l_0 state estimator into a convex program based on l_1/l_r optimization (e.g., $r = 2$) as done in [8]. In this case, providing a bound for the state-estimation error when the l_1/l_r convex relaxation is used will be an avenue for future work.

8. APPENDIX

Proof of Theorem 2. The proof uses a generalization of the proof of Theorem 2 from [14]. We start by assuming that there exist a non-vertex point $\tilde{\mathbf{x}}_0 = arg\max_{\mathbf{x}\in\mathbb{Q}} f(\mathbf{x})$, and that for every vertex point $\mathbf{x} \in \mathbb{Q}$, $f(\mathbf{x}) < f(\tilde{\mathbf{x}})$. Note that $\tilde{\mathbf{x}}_0$ exists since the polyhedron $\mathbb{Q}$ is bounded. Since $\tilde{\mathbf{x}}_0$ is not a vertex of $\mathbb{Q} \in \mathbb{R}^n$, less than n constraints used to specify $\mathbb{Q}$ are active at $\tilde{\mathbf{x}}_0$ [4]; we denote by d $(d < n)$ the number of active constraints in $\tilde{\mathbf{x}}_0$ and denote them as $\mathbf{a}_i^T\tilde{\mathbf{x}}_0 = b_i, i = \{j_1, ..., j_d\}$. Thus, the vectors $\mathbf{a}_i, i = j_1, ..., j_d$ belong to a proper subspace of $\mathbb{R}^n$ and there exists $\mathbf{d} \in \mathbb{R}^n$ such that $\mathbf{d} \neq \mathbf{0}$ and $\mathbf{a}_i^T\mathbf{d} = 0$, for all $i = j_1, ..., j_d$.

Consider vectors $\tilde{\mathbf{x}}_0 + \epsilon\mathbf{d}$ and $\tilde{\mathbf{x}}_0 - \epsilon\mathbf{d}$ for any $\epsilon > 0$. From the definition of $\tilde{\mathbf{x}}_0$, $f(\tilde{\mathbf{x}}_0) \geq f(\tilde{\mathbf{x}}_0 + \epsilon\mathbf{d})$ and $f(\tilde{\mathbf{x}}_0) \geq f(\tilde{\mathbf{x}}_0 - \epsilon\mathbf{d})$, meaning that

$$f(\tilde{\mathbf{x}}_0) \geq \frac{1}{2}(f(\tilde{\mathbf{x}}_0 + \epsilon\mathbf{d}) + f(\tilde{\mathbf{x}}_0 - \epsilon\mathbf{d})).$$

However, since f is a convex function, we have that for all $\epsilon > 0$

$$f(\tilde{\mathbf{x}}_0) = f(\tilde{\mathbf{x}}_0 + \epsilon\mathbf{d}) = f(\tilde{\mathbf{x}}_0 - \epsilon\mathbf{d}). \tag{29}$$

Because $\mathbb{Q}$ is bounded, there exists some $\epsilon_1 > 0$ for which additional constraint used to specify $\mathbb{Q}$ gets activated either at point $\tilde{\mathbf{x}}_0 + \epsilon_1\mathbf{d}$ or $\tilde{\mathbf{x}}_0 - \epsilon_1\mathbf{d}$. Let us denote that point as $\tilde{\mathbf{x}}_1$. Note that from (29), $f(\tilde{\mathbf{x}}_0) = f(\tilde{\mathbf{x}}_1)$ and that at $\tilde{\mathbf{x}}_1$ at least $d+1$ constraints used to specify $\mathbb{Q}$ are activate (because all previously active constrains will remain active).

The above procedure can be repeated until we reach a point $\tilde{\mathbf{x}}_r$ ($r \leq n - d$, i.e., in at most $n - d$ steps) in which n constraints are activated and where $f(\tilde{\mathbf{x}}_r) = f(\tilde{\mathbf{x}}_0)$. However, since in $\tilde{\mathbf{x}}_r$ exactly n constraints used to specify $\mathbf{Q}$ are active we have that $\tilde{\mathbf{x}}_r$ is a vertex of the polyhedron [4], and therefore there exist an optimal point of the maximization problem that is a vertex of the polyhedron $\mathbb{Q}$. □

9. REFERENCES

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

[2] "Spoofers" Use Fake GPS Signals to Knock a Yacht Off Course. MIT Technology Review, August 14, 2013.

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

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

[5] A. Cardenas, S. Amin, and S. S. Sastry. Research challenges for the security of control systems. In *Proc. 3rd USENIX Workshop on Hot topics in security*, 2008. Article 6.

[6] 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 Symposium*, 2011.

[7] J. P. Farwell and R. Rohozinski. Stuxnet and the future of cyber war. *Survival*, 53(1):23–40, 2011.

[8] H. Fawzi, P. Tabuada, and S. Diggavi. Secure state-estimation for dynamical systems under active adversaries. In *Proceedings of the 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 337–344, 2011.

[9] H. Fawzi, P. Tabuada, and S. Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. *arXiv preprint arXiv:1205.5073*, 2012.

[10] 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*, 95(1):138 – 162, 2007.

[11] 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)*, pages 447 –462, 2010.

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

[13] M. Pajic, N. Bezzo, J. Weimer, R. Alur, R. Mangharam, N. Michael, G. J. Pappas, O. Sokolsky, P. Tabuada, S. Weirich, et al. Towards synthesis of platform-aware attack-resilient control systems. In *Proc. 2nd ACM international conference on High Confidence Networked Systems*, pages 75–76, 2013.

[14] M. Pajic, R. Mangharam, O. Sokolsky, D. Arney, J. Goldman, and I. Lee. Model-driven safety analysis of closed-loop medical systems. *IEEE Transactions on Industrial Informatics*, 10(1):3–16, 2014.

[15] F. Pasqualetti, F. Dörfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, 2013.

[16] 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)*, volume 3, 2009.

[17] 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*, pages 55–72. Springer, 2013.

[18] J. Slay and M. Miller. Lessons learned from the maroochy water breach. In *Critical Infrastructure Protection*, pages 73–82, 2007.

[19] R. Smith. A decoupled feedback structure for covertly appropriating networked control systems. *Proc. IFAC World Congress*, pages 90–95, 2011.

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

[21] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson. Attack models and scenarios for networked control systems. In *Proc. 1st international conference on High Confidence Networked Systems*, HiCoNS'12, pages 55–64, 2012.

[22] J. S. Warner and R. G. Johnston. A simple demonstration that the global positioning system (gps) is vulnerable to spoofing. *Journal of Security Administration*, 25(2):19–27, 2002.

[23] W. Zhang, M. Branicky, and S. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, 2001.