Event-Triggered State Observers for Sparse Sensor Noise/Attacks

Yasser Shoukry and Paulo Tabuada

***Abstract*—This paper describes two algorithms for state reconstruction from sensor measurements that are corrupted with sparse, but otherwise arbitrary, "noise." These results are motivated by the need to secure cyber-physical systems against a malicious adversary that can arbitrarily corrupt sensor measurements. The first algorithm reconstructs the state from a batch of sensor measurements while the second algorithm is able to incorporate new measurements as they become available, in the spirit of a Luenberger observer. A distinguishing point of these algorithms is the use of event-triggered techniques to improve the computational performance of the proposed algorithms.**

***Index Terms*—Event-triggered observers, secure cyber-physical systems, secure state reconstruction, sensor attacks.**

I. Introduction

THE security of Cyber-Physical Systems (CPSs) has recently become a topic of scientific inquiry in no small part due to the discovery of the Stuxnet malware, the most famous example of an attack on process control systems [1]. Although one might be tempted to associate CPS security with large-scale and critical infrastructure, such as the power-grid or water distribution systems, previous work by the authors and co-workers has shown that even smaller systems, such as cars, can be attacked. It was shown in [2] how to attack the velocity measurements of anti-lock braking systems so as to force drivers to loose control of their vehicles.

In this paper, we propose two state observers for discrete-time linear systems in the presence of sparsely corrupted measurements. Sparse "noise" is a natural model to describe the effect of a malicious attacker that has the ability to alter the measurements of a subset of sensors in a feedback control loop. While measurements originating from un-attacked sensors are "noise" free, measurements from attacked sensors can be arbitrary: we make no assumption regarding its magnitude, statistical description, or temporal evolution. Hence, the noise vector is sparse; its elements are either zero or arbitrary real numbers.

Several results on state reconstruction under sensor attacks have recently appeared in the literature. We classify the existing work in two classes based on how the physical plant is modeled: 1) steady-state operation (no dynamics) and 2) linear time-invariant dynamics. In both classes the attacker is assumed to corrupt a few sensor measurements and thus its actions are adequately modeled as sparse noise.

The results reported in [3]–[7] fall in the first class—steady-state operation—and address security problems in the context of smart power grid systems. Due to the steady-state assumption, all of these results fail to exploit the constraints imposed by the continuous dynamics as done in this paper.

Representative work in the second class—linear time-invariant dynamics—includes [8]–[11]. The work reported in [8] addresses the detection of attacks through monitors inspired by the fault detection and isolation literature. Such methods are better suited for small systems since the number of monitors grows combinatorially with the number of sensors. The work reported in [9]–[11] draws inspiration from error correction over the reals [12] and compressive sensing [13] and formulates the secure state reconstruction problem as a $L_r \setminus L_1, r > 1$ optimization problem.

The problem of reconstructing the state under sensor attacks is closely related to fault-tolerant state reconstruction. The robust Kalman filter, described in [14], is the approach to fault-tolerant state reconstruction closer to the results in this paper, at the technical level. In robust Kalman filtering the state estimate updates are obtained by solving a convex L_1 optimization problem that is robust to outliers. With the advances in the computational power of current processors, the robust Kalman filter can be efficiently computed in real-time. However, no theoretical guarantees are known regarding the performance of this filter in the presence of malicious attacks.

In this paper, we extend the work in [9]–[11] by focusing on *efficient* algorithms in the sense of being implementable on computationally limited platforms. Rather than relying on classical algorithms for $L_r \setminus L_1, r > 1$ optimization, as was done in [9]–[11], we develop in this paper customized gradient-descent algorithms which have lightweight implementations. The computational efficiency claims are supported by numerical simulations showing an order of magnitude decrease in the computation time.

The proposed algorithms reconstruct both the state as well as the sparse noise/attack signal. Hence, they can be seen as an extension of compressive sensing techniques to the case where part of the signal to be reconstructed is sparse and the other part is governed by linear dynamics. A similar problem is studied in [15] where the recovery of a sparse streaming signal

Manuscript received May 25, 2015; revised September 3, 2015; accepted September 20, 2015. Date of publication October 27, 2015; date of current version July 22, 2016. This work was supported in part by the National Science Foundation Award 1136174 and by DARPA under agreement 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 NSF, DARPA or the U.S. Government. Recommended by Associate Editor H. Shim.

The authors are with the UCLA Electrical Engineering Department, Los Angeles, CA 90095-1594 USA (e-mail: yshoukry@ucla.edu; tabuada@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2015.2492159

with linear dynamics is discussed. Although the work in [15] also exploits the linear dynamics, it is not applicable to the state reconstruction problem where the sparse signal models an attack for which no dynamics is known.

Technically, we make the following contributions:

- The reconstruction or decoding of compressively sensed signals is characterized by properties such as the *restricted isometry* or the *restricted eigenvalues* [13]. We show that the relevant notion for state reconstruction under sensor attacks is a strong notion of observability introduced in this paper and termed s-sparse observability.
- We extend one of the previously proposed algorithms for the reconstruction or decoding of compressively sensed signals [16] to the case where part of the signal is sparse while the rest is governed by linear dynamics.
- We propose a recursive implementation of the method discussed in the previous bullet so that new measurement information can be used as it becomes available, in the spirit of a Luenberger observer.
- A distinguishing aspect of the proposed algorithms is the adoption of event-triggered techniques in order to enhance the computational performance.

A preliminary version of these results appeared in [17] where only the Luenberger observer (third bullet) was discussed and no proofs were given.

The rest of this paper is organized as follows. Section II formally introduces the problem under consideration. The notions of s-observability and s-restricted eigenvalues are introduced in Section III. The main results of this paper which are the Event-Triggered Projected Gradient-Descent algorithm and the Event-Triggered Projected Luenberger Observer, along with their convergence properties, are presented in Sections IV and V, respectively. Simulation results for the proposed algorithms are shown in Section VI. Finally, Section VII concludes this paper.

II. The Secure State Reconstruction Problem

A. Notation

The symbols $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}^+$ denote the set of natural, real, and positive real numbers, respectively. Given two vectors $x \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2}$, we denote by $(x, y) \in \mathbb{R}^{n_1+n_2}$ the vector $[x^T \quad y^T]^T$. We also use the notation z_x and z_y to denote the natural projection of the vector $z = (x, y)$ on its first and second component, respectively.

If S is a set, we denote by $|S|$ the cardinality of S. The support of a vector $x \in \mathbb{R}^n$, denoted by $\text{supp}(x)$, is the set of indices of the nonzero elements of x. We say that a vector $x \in \mathbb{R}^n$ is s-sparse, if x has at most s nonzero elements, i.e., if $|\text{supp}(x)| \leq s$.

Given p vectors of the same dimension $x_1, \ldots, x_p \in \mathbb{R}^n$, we call $x = (x_1, x_2, \ldots, x_p) \in \mathbb{R}^{pn}$ a block vector and each component x_i a block. To emphasize that a vector x is a block vector, we write it as an element of $\mathbb{R}^{pn}$ where the exponent pn is written as the juxtaposition of the number of blocks p and the size of individual blocks n, respectively. The block vector $x \in \mathbb{R}^{pn}$ is called block s-sparse, if, at most, s blocks are nonzero. For the sake of simplicity, we use s-sparse to denote block s-sparse. With some abuse of notation, for the block vector $x = (x_1, x_2, \ldots, x_p) \in \mathbb{R}^{pn}$ we denote by $\text{supp}(x)$ the indices of the blocks on which $x \in \mathbb{R}^{pn}$ is supported. In other words, an index $i \in \{1, \ldots, p\}$ belongs to the set $\text{supp}(x) \subseteq \{1, \ldots, p\}$ whenever the ith block x_i is nonzero, i.e.,

$$i \in \text{supp}(x) \Leftrightarrow x_i \neq 0, \quad i \in \{1, \ldots, p\}.$$

A block matrix $M \in \mathbb{R}^{pn \times m}$ is defined as the horizontal concatenation of m block vectors each of size pn. In such case, a block is defined as the matrix $M_i \in \mathbb{R}^{n \times m}$ and hence the matrix M can be written as $M = [M_1^T \ldots M_p^T]^T$. Note that similarly to the notation used for vectors, the row dimension of the block matrix $M \in \mathbb{R}^{pn \times m}$ is written as juxtaposition of the number of blocks p and the size of individual blocks n.

For a vector $x \in \mathbb{R}^n$, we denote by $\|x\|_2$ the 2-norm of x and by $\|M\|_2$ the induced 2-norm of a matrix $M \in \mathbb{R}^{m \times n}$. We also denote by $M_i \in \mathbb{R}^{1 \times n}$ the ith row of M. For a set $\Gamma \subseteq \{1, \ldots, m\}$, we denote by $M_\Gamma \in \mathbb{R}^{|\Gamma| \times n}$ the matrix obtained from M by removing all the rows except those indexed by Γ. We also denote by $M_{\overline{\Gamma}} \in \mathbb{R}^{(m-|\Gamma|) \times n}$ the matrix obtained from M by removing the rows indexed by the set Γ, for example, if $m = 4$, and $\Gamma = \{1, 2\}$, then:

$$M_\Gamma = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \quad \text{and} \quad M_{\overline{\Gamma}} = \begin{bmatrix} M_3 \\ M_4 \end{bmatrix}.$$

Using the same abuse of notation, for a block matrix $M \in \mathbb{R}^{pn \times m}$, we denote by $M_\Gamma \in \mathbb{R}^{|\Gamma| n \times m}$ the block matrix obtained by removing all blocks except those indexed by Γ. We define $M_{\overline{\Gamma}}$ similarly.

Finally, we denote the minimum eigenvalue and the maximum eigenvalue of a symmetric matrix M by $\lambda_{\min}\{M\}$ and $\lambda_{\max}\{M\}$ respectively.

B. Dynamics and Attack Model

Consider the following linear discrete-time control system where $x(t) \in \mathbb{R}^n$ is the system state at time $t \in \mathbb{N}$, $u(t) \in \mathbb{R}^m$ is the system input, and $y(t) \in \mathbb{R}^p$ is the observed measurement:

$$x(t+1) = Ax(t) + Bu(t) \tag{1}$$
$$y(t) = Cx(t) + a(t). \tag{2}$$

The matrices A, B, and C have appropriate dimensions and $a(t) \in \mathbb{R}^p$ is an s-sparse vector modeling how an attacker changed the sensor measurements at time t. If sensor $i \in \{1, \ldots, p\}$ is attacked then the ith element in the vector $a(t)$ is nonzero otherwise the ith sensor is not attacked. Hence, s describes the number of attacked sensors. Note that we make no assumptions on the vector $a(t)$ other than being s-sparse. In particular, we do not assume bounds, statistical properties, nor restrictions on the time evolution of the elements in $a(t)$. The value of s is also not assumed to be known although we assume the knowledge of an upper bound. The set of sensors the attacker has access to is assumed to remain constant over time (and has cardinality at most s). However, the attacker has complete freedom in deciding which sensor or sensors in this set are attacked and when, including the possibility of attacking all of them at all times.

Our objective is to simultaneously construct a delayed version of the state, $x(t-\tau+1)$, and the attack vectors $a(t-\tau+1), a(t-\tau+2), \ldots, a(t)$ from the measurements $y(t-\tau+1), y(t-\tau+2), \ldots, y(t)$.

By collecting $\tau \in \mathbb{N}$ observations (starting from time $t \geq \tau$) with $\tau \leq n$, we can write the output from the ith sensor as follows:

$$\widetilde{Y}_i(t) = \mathcal{O}_i x(t-\tau+1) + E_i(t) + F_i U(t) \tag{3}$$

where:

$$\widetilde{Y}_i(t) = \begin{bmatrix} y_i(t-\tau+1) \\ y_i(t-\tau+2) \\ \vdots \\ y_i(t) \end{bmatrix}, \quad E_i(t) = \begin{bmatrix} a_i(t-\tau+1) \\ a_i(t-\tau+2) \\ \vdots \\ a_i(t) \end{bmatrix}$$

$$U(t) = \begin{bmatrix} u(t-\tau+1) \\ u(t-\tau+2) \\ \vdots \\ u(t) \end{bmatrix}, \quad \mathcal{O}_i = \begin{bmatrix} C_i \\ C_i A \\ \vdots \\ C_i A^{\tau-1} \end{bmatrix}$$

$$F_i = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ C_i B & 0 & \dots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ C_i A^{\tau-2} B & C_i A^{\tau-3} B & \dots & C_i B & 0 \end{bmatrix}.$$

Since all the inputs in $U(t)$ are known, we can further simplify (3) as

$$Y_i(t) = \mathcal{O}_i x(t-\tau+1) + E_i(t) \tag{4}$$

where $Y_i(t) = \widetilde{Y}_i(t) - F_i U(t)$. We also define

$$\widetilde{Y}(t) = \begin{bmatrix} \widetilde{Y}_1(t) \\ \vdots \\ \widetilde{Y}_p(t) \end{bmatrix}, \quad Y(t) = \begin{bmatrix} Y_1(t) \\ \vdots \\ Y_p(t) \end{bmatrix}$$

$$E(t) = \begin{bmatrix} E_1(t) \\ \vdots \\ E_p(t) \end{bmatrix}, \quad \mathcal{O} = \begin{bmatrix} \mathcal{O}_1 \\ \vdots \\ \mathcal{O}_p \end{bmatrix}. \tag{5}$$

Here, the vectors $\widetilde{Y}_i, Y_i, E_i \in \mathbb{R}^\tau$ and the matrix $\mathcal{O}_i \in \mathbb{R}^{\tau \times n}$ are used to denote the ith block of $\widetilde{Y}, Y, E \in \mathbb{R}^{p\tau}$, and $\mathcal{O} \in \mathbb{R}^{p\tau \times n}$, respectively. Now, we can write the output equation as

$$\begin{aligned} Y(t) &= \mathcal{O} x(t-\tau+1) + E(t) \\ &= [\mathcal{O} \quad I] \begin{bmatrix} x(t-\tau+1) \\ E(t) \end{bmatrix} = Q z(t) \end{aligned} \tag{6}$$

where $z(t) = (x(t-\tau+1), E(t))$ and $Q = [\mathcal{O} \quad I]$.

It is worth explaining the block sparse nature of $E(t)$ in more detail. Consider the attack vectors $a(t-\tau+1), a(t-\tau+2), \dots, a(t)$ and reformat this data as the matrix $\widetilde{E}(t) \in \mathbb{R}^{p \times \tau}$ where column j is given by $a(t-j+1)$. Since we assume that the set of sensors under attack does not change over time, the sparsity pattern appears in $\widetilde{E}(t)$. The rows corresponding to the unattacked sensors have zeros only, while the rows corresponding to the attacked sensors have arbitrary (zero or nonzero) elements.

Example 2.1: Consider a system with four sensors, $\tau = 4$, and an attack on the second and third sensors. The attack matrix $\widetilde{E}(t)$ will be of the form

$$\widetilde{E}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & 5 & 6 & 10 \\ 4 & 8 & 0 & 12 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The block sparsity structure appears once the attack matrix $\widetilde{E}(t)$ is reshaped as the vector $E(t)$

$$E(t) = [0 \quad 0 \quad 0 \quad 0 \quad 2 \quad 5 \quad 6 \quad 10 \quad 4 \quad \dots]^T.$$

Since block s-sparse vectors pervade this paper, it is convenient to denote them by a special symbol.

Definition 2.2 (Block s-Sparse Set $\mathbb{S}_s$): The subset of $\mathbb{R}^{p\tau}$ consisting of the block vectors that are block s-sparse is denoted by $\mathbb{S}_s$.

Using this block sparsity notion, we pose two problems which will lead to the two novel algorithms for state reconstruction under sensor attacks.

C. Static Batch Optimization

Problem 2.3 (Static Batch Optimization): For the linear control system defined by (1) and (2) construct the estimate $\hat{z} = (\hat{x}, \hat{E})$, where $\hat{x} \in \mathbb{R}^n$ is the state estimate and $\hat{E} \in \mathbb{S}_s$ is the attack vector estimate, obtained as the solution of the following optimization problem:

$$\arg \min_{\hat{z} \in \mathbb{R}^n \times \mathbb{S}_s} \frac{1}{2} \|Y - Q\hat{z}\|_2^2.$$

We dropped the time t argument since this optimization problem is to be solved at every time instance. We note that we seek a solution in the nonconvex set $\mathbb{R}^n \times \mathbb{S}_s$ and no closed-form solution is known for this problem. We also note that this optimization problem asks for the reconstruction of a delayed version of the state $x(t-\tau+1)$. However, we can always reconstruct of the current state $x(t)$ from $x(t-\tau+1)$ by recursively rolling the dynamics forward in time. Alternatively, when A is invertible, we can directly recover $x(t)$ by re-writing the measurement equation as a function of $x(t)$.

As a final remark we note that it follows from the Cayley–Hamilton theorem that there is no loss of generality in taking the number of collected measurements τ to be no greater than the number of states n.

D. Luenberger-Like Observer

Problem 2.3 asks for the reconstruction of $z(t)$ using a batch approach based on the measured data in the vector Y. On computationally restricted platforms we may be faced with the difficulty of having to process new measurements before being able to compute a solution to Problem 2.3. It would then be preferable to reconstruct $z(t)$ using an algorithm that can incorporate new measurements as they become available. This motivates the following problem.

Problem 2.4 (Luenberger-Like Observer): For the linear control system defined by (1) and (2) construct a dynamical system

$$\hat{z}(t+1) = f\left(\hat{z}(t), U(t), \widetilde{Y}(t)\right)$$

such that

$$\lim_{t \to \infty} (z^*(t) - \hat{z}(t)) = 0$$

where $z^*(t) = (x^*(t-\tau+1), E^*(t)) \in \mathbb{R}^n \times \mathbb{S}_s$, $x^*(t)$ is the solution of (1) under the inputs $U(t)$, and $\widetilde{Y}(t)$ is the sequence of the last τ observed outputs corrupted by $E^*(t)$.

III. s-Sparse Observability and the Restricted Eigenvalue

Recall that Problem 2.3 asks for the minimizer of $(1/2)\|Y - Q\hat{z}\|_2^2$. Since the matrix Q has a nontrivial kernel, there exist many pairs $\hat{z} = (\hat{x}, \hat{E})$ which solve this problem. In this section, we look closely at the fundamental question of uniqueness of solutions.

A. s-Sparse Observability

We start by introducing the following notion of observability.

Definition 3.1 (s-Sparse Observable System): The linear control system defined by (1) and (2) is said to be s-sparse observable if for every set $\Gamma \subseteq \{1, \ldots, p\}$ with $|\Gamma| = s$, the pair $(A, C_{\overline{\Gamma}})$ is observable.

A system is s-sparse observable if it remains observable after eliminating any choice of s sensors. This notion of observability underlies most of the results in this paper and can also be expressed using the notion of strong observability for linear systems described in [18], [19]. To make this connection explicit, consider a linear system $(A;\ B = 0;\ C;\ D = I_a)$ where the attack signal $a(t)$ is regarded as an input and I_a is the diagonal matrix defined so that the ith element on the diagonal is zero whenever $a_i(t)$ is zero and is one otherwise. Since in our formulation we assume no knowledge of the support of the attack signal $a(t)$, we need to consider all the different supports for the attack vector. It is then not difficult to see that a linear system is s-sparse observable if and only if the systems $(A; B = 0; C; D = I_a)$ are strong observable for all the matrices I_a obtained by considering attack vectors with all the possible supports.

We use the notion of s-sparse observability to characterize the uniqueness of solutions to Problem 2.3.

Theorem 3.2 (Existence and Uniqueness of Solutions to Problem 2.3): Problem 2.3 has a unique solution, i.e., the function $(1/2)\|Y - Q\hat{z}\|_2^2$ has a unique minimum on the set $\mathbb{R}^n \times \mathbb{S}_s$, if and only if the linear dynamical system defined by (1) and (2) is $2s$-sparse observable.

Proof: We first note that $\|Y - Q\hat{z}\|_2^2$ is always nonnegative and it becomes zero whenever $\hat{x}$ is the true state and $\hat{E}$ is the true attack vector. Hence, $\|Y - \mathcal{O}\hat{x} - \hat{E}\|_2^2$ has a unique minimum if and only if the equality $Y = Q\hat{z} = \mathcal{O}\hat{x} + \hat{E}$ only holds for the true state and the true attack vector. However, this is equivalent to injectivity of the map $f : \mathbb{R}^n \times \mathbb{S}_s \to \mathbb{R}^{p\tau}$ defined by $f(x, E) = \mathcal{O}x + E$.

To prove the stated result we assume, for the sake of contradiction, that f is not injective and (A, C) is $2s$-sparse observable. Since f is not injective there exist $(x, E), (x', E') \in \mathbb{R}^n \times \mathbb{S}_s$ with $(x, E) \neq (x', E')$ such that $f(x, E) = f(x', E')$. We now note that

$$f(x, E) = f(x', E') \Leftrightarrow \mathcal{O}x + E = \mathcal{O}x' + E' \Leftrightarrow \mathcal{O}(x - x') = E' - E.$$

Let $\Gamma = \text{supp}(E' - E)$ be the set consisting of the indices for the blocks where $E' - E$ is supported. It follows from E and E' being block s-sparse that the support $|\Gamma|$ is at most $2s$. Then, $\mathcal{O}(x - x') = E' - E$ implies $\mathcal{O}_{\overline{\Gamma}}(x - x') = 0$ which in turn implies that the pair $(A, C_{\overline{\Gamma}})$ is not observable since $x \neq x'$, a contradiction.

Conversely, if the pair $(A, C_{\overline{\Gamma}})$ is not observable while $|\Gamma|$ is at most $2s$, then there exists a nonzero vector v such that $v \in \ker \mathcal{O}_{\overline{\Gamma}}$. Fix $x \in \mathbb{R}^n$, let $x' = x + v$, and note that $\mathcal{O}_{\overline{\Gamma}}(x - x') = 0$. Split Γ in two sets Γ_1 and Γ_2 so that $\Gamma = \Gamma_1 \cup \Gamma_2$, $|\Gamma_1| \leq s$ and $|\Gamma_2| \leq s$. We can now define E by making its ith block E_i to be equal to $\mathcal{O}_i(x - x')$ if $i \in \Gamma_1$ and zero otherwise. Similarly, we define E' by making its ith block E'_i to be equal to $\mathcal{O}_i(x' - x)$ if $i \in \Gamma_2$ and zero otherwise. As defined, E and E' are block s-sparse and the equality $f(x, E) = f(x', E')$ holds thereby showing that f is not injective. ■

Although the notion of s-sparse observability is of combinatorial nature, since we have to check observability of all possible pairs $(A, C_{\overline{\Gamma}})$, it does clearly illustrate a fundamental limitation: it is impossible to correctly reconstruct the state whenever $p/2$ or more sensors are attacked. Indeed, suppose that we have an even number of sensors p and $s = p/2$ sensors are attacked. Theorem 3.2 requires the system to still be observable after removing $2s = p$ rows from the matrix C which leads to $C_{\overline{\Gamma}}$ being the null matrix. This fundamental limitation is consistent with and had been previously reported in [9], [10], [19].

In [9], [10], the possibility of reconstructing the state under sensor attacks is characterized by Proposition 2 and, under certain assumptions on the A matrix, by Lemma 1. The characterization based on s-sparse observability complements the characterizations in [9], [10] in the following sense. Proposition 2 in [9], [10] requires a test to be performed for every state $x \in \mathbb{R}^n$ and does not lead to an effective algorithm. In contrast, s-sparse observability only requires a finite, albeit large, number of computations. Lemma 1 in [9] requires an even smaller number of tests than s-sparse observability but only applies under additional assumptions on the A matrix. Moreover, s-sparse observability connects the state reconstruction problem under sensor attacks to the well known systems theoretic notion of observability.

B. s-Restricted Eigenvalue

While s-sparse observability provides a qualitative characterization of the existence and uniqueness of solutions to Problem 2.3, in this section we discuss a more quantitative version termed the *s-restricted eigenvalue property* [20]. This property is directly related to the possibility of solving Problem 2.3 and Problem 2.4 using gradient descent inspired methods.

Definition 3.3 (s-Restricted Eigenvalue of a Linear Control System): For a given set $\Gamma_s \subseteq \{1, \ldots, p\}$ with $|\Gamma_s| = s$, (with some abuse of notation) the matrix $Q_{\Gamma_s} \in \mathbb{R}^{p\tau \times (n + s\tau)}$ is defined by

$$Q_{\Gamma_s} = \begin{bmatrix} \mathcal{O} & (I_{\Gamma_s})^T \end{bmatrix} \tag{7}$$

where $I \in \mathbb{R}^{p\tau \times p\tau}$ is the identity matrix. The s-restricted eigenvalue of the control system defined by (1) and (2) is the smallest eigenvalue of all the matrices $Q_{\Gamma_s}^T Q_{\Gamma_s}$ obtained by considering all the different sets Γ_s.

The s-restricted eigenvalue of a control system can be related to s-observability as follows.

Proposition 3.4 (Nonzero Restricted Eigenvalue): Let the linear control system, defined by (1) and (2), be $2s$-sparse observable. There exists a $\delta_{2s} \in \mathbb{R}^+$ such that the $2s$-restricted eigenvalue is no smaller than δ_{2s}.

In other words, although Q has a nontrivial kernel, Q^TQ has a nonzero minimum eigenvalue when operating on a vector $z = (x, E)$ with block $2s$-sparse z_E, i.e., the following inequality holds for $z = (x, E) \in \mathbb{R}^n \times \mathbb{S}_{2s}$:

$$0 < \delta_{2s} z^T z \leq z^T Q^T Q z. \tag{8}$$

Proof: For any vector $z = (x, E) \in \mathbb{R}^n \times \mathbb{S}_{2s}$ we define Γ_{2s} and $z_{\Gamma_{2s}}$ as

$$\Gamma_{2s} = \text{supp}(z_E), \quad z_{\Gamma_{2s}} = (x, E_{\Gamma_{2s}}).$$

Since $z^TQ^TQz = z_{\Gamma_{2s}}^T Q_{\Gamma_{2s}}^T Q_{\Gamma_{2s}} z_{\Gamma_{2s}}$, we have

$$\begin{aligned} z^TQ^TQz &= z_{\Gamma_{2s}}^T Q_{\Gamma_{2s}}^T Q_{\Gamma_{2s}} z_{\Gamma_{2s}} \\ &\geq \lambda_{\min}\left\{Q_{\Gamma_{2s}}^T Q_{\Gamma_{2s}}\right\} z_{\Gamma_{2s}}^T z_{\Gamma_{2s}} \geq \delta_{2s} z^T z \end{aligned}$$

where the last inequality follows from the fact that $z^Tz = z_{\Gamma_{2s}}^T z_{\Gamma_{2s}}$ and by defining δ_{2s} as

$$\delta_{2s} = \min_{\Gamma_{2s}} \quad \lambda_{\min}\left\{Q_{\Gamma_{2s}}^T Q_{\Gamma_{2s}}\right\}.$$

To conclude that inequality (8) holds, we need to show that δ_{2s} is strictly positive. This follows from the fact that the lower bound $\delta_{2s} z^T z$ is achieved by choosing $z_{\Gamma_{2s}}$ to be the eigenvector of $Q_{\Gamma_{2s}}^T Q_{\Gamma_{2s}}$ associated with the eigenvalue δ_{2s}. It then follows from Theorem 3.2 that δ_{2s} must be strictly positive since $z^TQ^TQz \neq 0$ for any $z = (x, E) \neq 0$ with z_E block $2s$-sparse. ■

Similarly to $2s$-sparse observability, the computation of δ_{2s} is combinatorial in nature since one needs to calculate the eigenvalues of $Q_{\Gamma_{2s}}^T Q_{\Gamma_{2s}}$ for all the different sets Γ_{2s}.

IV. Event-Triggered Projected Gradient Descent

The problem of reconstructing a sparse signal from measurements is well studied in the compressive sensing literature [21], [22]. In this section, we present an algorithm for state reconstruction that can be seen as an extension of the iterative hard thresholding algorithm, reported in [16], to the case where part of the signal to be reconstructed is sparse and part is governed by linear dynamics. We also draw inspiration from [23] where it is shown how to interpret optimization algorithms as dynamical systems with feedback. Once this link is established, Lyapunov analysis techniques become available to design as well as to analyze the performance of optimization algorithms.

A. The Algorithm

The Event-Triggered Projected Gradient Descent (ETPG) algorithm updates the estimate $\hat{z} \in \mathbb{R}^n \times \mathbb{R}^{p\tau}$ of $z \in \mathbb{R}^n \times \mathbb{S}_s$ by following the gradient direction of the Lyapunov candidate function $V(\hat{z}) = (1/2)\|Y - Q\hat{z}\|_2^2$, i.e.,

$$\hat{z}^+ := \hat{z} + \eta Q^T (Y - Q\hat{z})$$

for some step size $\eta \in \mathbb{R}^+$. Since $\hat{z}$ will not, in general, satisfy the desired sparsity constraints, gradient steps are alternated with projection steps using the projection operator

$$\Pi : \mathbb{R}^n \times \mathbb{R}^{p\tau} \to \mathbb{R}^n \times \mathbb{S}_s$$

that takes $\hat{z} \in \mathbb{R}^n \times \mathbb{R}^{p\tau}$ to the closest point in $\mathbb{R}^n \times \mathbb{S}_s$. In order to ensure that V decreases along the execution of the algorithm, multiple gradient steps are performed for each projection step. By monitoring the evolution of V, the algorithm can determine when the decrease in V caused by the gradient steps compensates the increase caused by Π. This is akin to triggering an event in event-triggered control [24].

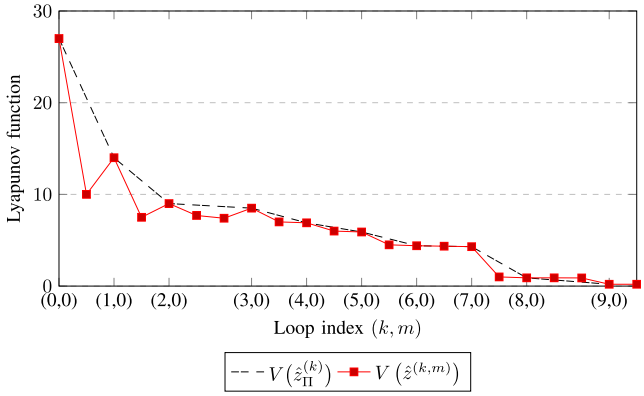


Fig. 1. An example of the evolution of the Lyapunov candidate function V while running the ETPG algorithm. The subsequence highlighted in blue (dashed) is a decreasing subsequence.

Let z^* denotes the solution of Problem 2.3 and note that the error $z_E^* - \hat{z}_E$ is, at most, $2s$-sparse whenever $\hat{z}_E$ is at most s-sparse. From Theorem 3.2 we know that the intersection of the set $\mathbb{R}^n \times \mathbb{S}_{2s}$ with the kernel of Q is only one point which corresponds to $\hat{z} = z^*$. Hence, by driving $V(\hat{z})$ to zero while forcing $\hat{z}_E$ to be s-sparse, $\hat{z}$ is guaranteed to converge to z^*. Formalizing these ideas leads to Algorithm 1.

Algorithm 1 Event-Triggered Projected Gradient Descent

Initialize $k := 1$, $m := 0$, $\hat{z}_\Pi^{(0)} := 0$;
while $V(\hat{z}_\Pi^{(k-1)}) \geq \alpha$ **do**
  $\hat{z}^{(k,0)} := \hat{z}_\Pi^{(k-1)}$;
  reset $m := 0$, $V_{\text{temp}} := V(\hat{z}_\Pi^{(k-1)})$;
  while $V_{\text{temp}} \geq (1-\nu)V(\hat{z}_\Pi^{(k-1)})$ **do**
    $\hat{z}^{(k,m+1)} := \hat{z}^{(k,m)} + \eta Q^T(Y - Q\hat{z}^{(k,m)})$;
    $V_{\text{temp}} := V(\Pi(\hat{z}^{(k,m+1)}))$;
    $m := m + 1$;
  end while
  $\hat{z}_\Pi^{(k)} := \Pi(\hat{z}^{(k,m)})$;
  $k := k + 1$;
end while
return $\hat{z}_\Pi^{(k)}$

A typical execution of the ETPG algorithm is illustrated in Fig. 1 where the evolution of the Lyapunov candidate function V is shown. We can see that at $k = 0, 1$ the ETPG algorithm applied one gradient step before applying the new projection step, while at $k = 2$ and $k = 8$ two gradient steps were executed to compensate for the increase in V caused by the projection step. This adaptive nature of the ETPG algorithm is a consequence of the event-triggering mechanism. The algorithm ensures that the subsequence depicted in blue in Fig. 1 is a converging subsequence.

The ETPG algorithm uses two counters, one for each loop. Accordingly, we will use the notation $\hat{z}^{(k,m)}$ where k counts the number of iterations of the outer loop, while m counts the number of internal gradient descent steps within each iteration of the inner loop.

B. The Projection Operator

Before discussing the convergence of the proposed algorithm, it is important to show how to compute the projection operator Π.

Definition 4.1: Given a vector $z = (x, E) \in \mathbb{R}^n \times \mathbb{R}^{p\tau}$, we denote by $\Pi(z)$ the element of $\mathbb{R}^n \times \mathbb{S}_s$ closest to z in the 2-norm sense, i.e.,

$$\|\Pi(z) - z\|_2 \leq \|z' - z\|_2 \tag{9}$$

for any $z' \in \mathbb{R}^n \times \mathbb{S}_s$.

We first note that $\Pi(z) = \Pi(x, E) = (x, \Pi'(E))$. To explain how Π' is computed, recall from Example 2.1 the matrix $\widetilde{E}$, obtained by formatting $E \in \mathbb{R}^{p\tau}$ so that the ith column of $\widetilde{E}$ is given by $a(t - i + 1)$. Define $\overline{E} \in \mathbb{R}^p$ by $\overline{E}_i = \|\widetilde{E}_i\|_2$. By noting that $\|E\|_2^2 = \|\overline{E}\|_2^2$ and that E is block s-sparse if and only if $\overline{E}$ is s-sparse, it immediately follows that $\Pi'(z)$ can be computed by setting to zero the elements of E corresponding to the smallest $p - s$ entries of $\overline{E}$.

Example 4.2: Let us consider the following example with $p = 3$, $\tau = 3$ and $s = 1$:

$$E = [1 \quad 4 \quad 7 \quad 2 \quad 5 \quad 8 \quad 3 \quad 6 \quad 9]^T \in \mathbb{R}^{p\tau}.$$

Hence

$$\widetilde{E} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \in \mathbb{R}^{p\times\tau}, \quad \overline{E} = \begin{bmatrix} 8.164 \\ 9.6437 \\ 11.225 \end{bmatrix} \in \mathbb{R}^p$$

$$\Pi'(\overline{E}) = \begin{bmatrix} 0 \\ 0 \\ 11.225 \end{bmatrix} \in \mathbb{R}^p$$

which leads to

$$\Pi'(E) = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 3 \quad 6 \quad 9]^T.$$

Remark 4.3: Since the set $\mathbb{R}^n \times \mathbb{S}_s$ is nonconvex, the projection operator Π may not be well defined at each point. Whenever there is more than one point minimizing the distance to z, we define the projection as being one of these minimizers. None of the results in this paper depend on the choice of the minimizer, they only depend on the inequality (9).

C. Convergence of the ETPG Algorithm

In this subsection, we discuss the convergence and performance of the ETPG algorithm. The main result of this section is stated in the following theorem.

Theorem 4.4 (Convergence of the ETPG Algorithm): Let the linear control system defined by (1) and (2) be $2s$-sparse observable with $2s$-restricted eigenvalue δ_{2s} and let $z^* = (x^*, E^*) \in \mathbb{R}^n \times \mathbb{S}_s$ denote the solution of Problem 2.3. If the following holds:

1) $0 < \eta < \lambda_{\max}^{-1}\{Q^TQ\}$;
2) $\delta_{2s} > (4/9)\lambda_{\max}\{Q^TQ\}$;

then, for any error tolerance $\varepsilon \in \mathbb{R}^+$, there exists a termination constant $\alpha \in \mathbb{R}^+$, a number $k \in \mathbb{N}$ of outer loop iterations, and an inner loop triggering constant ν such that the estimate $\hat{z}_\Pi^{(k)}$ computed by the ETPG algorithm during its kth iteration satisfies $\|z^* - \hat{z}_\Pi^{(k)}\|_2 \leq \varepsilon$.

Remark 4.5: Theorem 4.4 shows that the ETPG algorithm correctly reconstructs the state, up to a desired error ε, whenever the attacker has access to no more than s sensors, the system is $2s$-sparse observable, and condition 2) on the restricted eigenvalue is satisfied. In practice, one does not exactly know the number s of attacked sensors. However, if an upper bound $\overline{s}$ is known, the ETPG is guaranteed to work as long as the system is $2\overline{s}$-sparse observable. For this reason the ETPG algorithm is typically executed with the largest s for which $2s$-spare observability holds.

Remark 4.6: The ETPG algorithm can be seen as a generalization of the "Normalized Iterative Hard Thresholding" (NIHT) algorithm presented in [25]. The ETPG algorithm uses multiple gradient descent steps supervised by an event-triggering mechanism whereas the NIHT algorithm only uses one gradient step. These two ingredients, multiple gradient steps and event-triggering, result in weaker conditions for convergence: $\delta_{2s} > (4/9)\lambda_{\max}\{Q^TQ\}$ for the ETPG algorithm vs $\delta_{2s} > (8/9)\lambda_{\max}\{Q^TQ\}$ for the NIHT algorithm (Theorem 4 in [25]).

In the remainder of this subsection, we focus on proving this theorem by resorting to the Lyapunov candidate function

$$W(\hat{z}) = \|z^* - \hat{z}\|_2 = \|e\|_2.$$

Note that, unlike V, the Lyapunov candidate function W has a unique minimum at $\hat{z} = z^*$. However, one should note that evaluation of W requires the knowledge of z^* which we do not have *a priori*. Accordingly, W is only adequate to discuss stability but can not be used to design the ETPG algorithm.

The proof is divided into small steps. First we discuss the effect of the two operations used inside the ETPG algorithm: projection and gradient descent. This is done in Propositions 4.7, 4.8, and 4.9. Next, we provide sufficient conditions under which the inner loop is guaranteed to terminate. These are given in Proposition 4.11. Finally, we use the triggering condition to establish that W is a Lyapunov function.

1) Effect of the Projection Operator:

Proposition 4.7: The following inequality holds for any $z \in \mathbb{R}^n \times \mathbb{R}^{p\tau}$:

$$W \circ \Pi(z) \leq 2W(z).$$

Proof: The result is proved by direct computation as follows:

$$\begin{aligned} W \circ \Pi(z) = \|z^* - \Pi(z)\|_2 &\leq \|z^* - z\|_2 + \|\Pi(z) - z\|_2 \\ &\leq 2\|z^* - z\|_2 = 2W(z) \end{aligned}$$

where the first inequality follows from the triangular inequality and the second inequality follows from (9). ∎

2) Effect of Gradient Steps: In this subsection, we study the effect of the gradient descent step along with sufficient conditions for termination of the inner loop of the ETPG algorithm. The first proposition is a technical result used in the proof of the second proposition that characterizes the decease in W caused by gradient descent steps.

Proposition 4.8: Let the linear control system defined by (1) and (2) be $2s$-sparse observable with $2s$-restricted eigenvalue δ_{2s}. The following holds for any $z \in \mathbb{R}^n \times \mathbb{S}_{2s}$:

$$\left\|(I - Q^+Q)z\right\|_2 \leq \left(1 - \delta_{2s}\lambda^{-1}_{\max}\{Q^TQ\}\right)^{\frac{1}{2}} \|z\|_2$$

where Q^+ is the Moore-Penrose pseudo inverse of Q.

Proof: The result follows from direct computations as follows:

$$\begin{aligned}\left\|(I - Q^+Q)z\right\|_2^2 &= z^T(I - Q^+Q)^2 z \overset{(a)}{=} z^T(I - Q^+Q)z \\ &\overset{(b)}{=} z^T\left(I - Q^T(QQ^T)^{-1}Q\right)z \\ &\overset{(c)}{\leq} z^T\left(I - \lambda^{-1}_{\max}\{Q^TQ\}Q^TQ\right)z \\ &\overset{(d)}{\leq} \left(1 - \delta_{2s}\lambda^{-1}_{\max}\{Q^TQ\}\right)z^Tz\end{aligned}$$

where equality (a) follows by noticing that the projection operator $I - Q^+Q$ is idempotent; (b) follows from the definition of $Q^+ = Q^T(QQ^T)^{-1}$; (c) follows from the fact that $QQ^T = I + \mathcal{O}O^T$ is a positive definite matrix and hence the inverse $(QQ^T)^{-1}$ exists and can be bounded as $(QQ^T)^{-1} \geq \lambda^{-1}_{\max}\{Q^TQ\}I$; and (d) follow from the $2s$-sparse observability assumption along with Proposition 3.4. ■

Proposition 4.9: Let the linear control system defined by (1) and (2) be $2s$-sparse observable with $2s$-restricted eigenvalue δ_{2s}. If the following conditions hold:

1) the estimate $\hat{z}_E^{(k,0)}$ is s-sparse, i.e., $\hat{z}^{(k,0)} = (\hat{z}_x^{(k,0)}, \hat{z}_E^{(k,0)}) \in \mathbb{R}^n \times \mathbb{S}_s$,
2) the step size $\eta \in \mathbb{R}^+$ satisfies $\eta < \lambda^{-1}_{\max}\{Q^TQ\}$, then for any $\xi \in]0,1[$, the following inequality holds for any $m \geq \lceil \log\xi / \log(1 - \eta\delta_{2s})\rceil$:

$$W\left(\hat{z}^{(k,m)}\right) \leq \left(\left(1 - \delta_{2s}\lambda^{-1}_{\max}\{Q^TQ\}\right)^{\frac{1}{2}} + \xi\right) W\left(\hat{z}^{(k,0)}\right) \tag{10}$$

where $\hat{z}^{(k,m)} \in \mathbb{R}^n \times \mathbb{R}^{p\tau}$ is recursively defined by

$$\hat{z}^{(k,m+1)} := \hat{z}^{(k,m)} + \eta Q^T\left(Y - Q\hat{z}^{(k,m)}\right).$$

Proof: The Lyapunov candidate function $W(\hat{z}^{(k,m)})$, after taking one gradient descent step, can be written as follows:

$$\begin{aligned}W\left(\hat{z}^{(k,m)}\right) &= \left\|z^* - \hat{z}^{(k,m)}\right\|_2 \\ &= \left\|z^* - \hat{z}^{(k,m-1)} - \eta Q^T\left(Y - Q\hat{z}^{(k,m-1)}\right)\right\|_2 \\ &= \left\|z^* - \hat{z}^{(k,m-1)} - \eta Q^T\left(Qz^* - Q\hat{z}^{(k,m-1)}\right)\right\|_2 \\ &= \left\|e^{(k,m-1)} - \eta Q^TQe^{(k,m-1)}\right\|_2.\end{aligned}$$

Recursively extending the previous analysis to m steps, we obtain

$$W\left(\hat{z}^{(k,m)}\right) = \left\|(I - \eta Q^TQ)^m e^{(k,0)}\right\|_2. \tag{11}$$

Now define the projection error $\Delta(m)$ as

$$\Delta(m) = (I - Q^+Q) - (I - \eta Q^TQ)^m \tag{12}$$

where Q^+ is the Moore-Penrose pseudo inverse of Q. It follows from Corollary 3 in [26] that the matrix $(I - \eta Q^TQ)^m$ converges to $I - Q^+Q$ whenever the the eigenvalues of the matrix $I - \eta Q^TQ$ lies inside the unit circle. However, this condition on the eigenvalues of the matrix $I - \eta Q^TQ$ follows directly from assumption 2) on the step size η. In this case, Corollary 3 in [26] states that the projection error $\Delta(m)$ is bounded from above as:

$$\|\Delta(m)\|_2 \leq (1 - \eta\delta_{2s})^m.$$

Note that $1 - \eta\delta_{2s}$ satisfies $0 < 1 - \eta\delta_{2s} < 1$. This can be shown as follows. First, we note that the ratio $\delta_{2s}\lambda^{-1}_{\max}\{Q^TQ\}$ satisfies $\delta_{2s}\lambda^{-1}_{\max}\{Q^TQ\} \leq 1$. This follows from δ_{2s} being defined as the smallest eigenvalue of Q^TQ when it operates on vectors belonging to the subset $\mathbb{R}^n \times \mathbb{S}_{2s}$ while $\lambda_{\max}\{Q^TQ\}$ is the maximum eigenvalue of the same matrix without any restrictions. Combining the bound $\delta_{2s}\lambda^{-1}_{\max}\{Q^TQ\} \leq 1$ along with the assumption 2) on the step size η, we conclude that $1 - \eta\delta_{2s}$ satisfies $0 < 1 - \eta\delta_{2s} < 1$. Now, given any $\xi \in]0,1[$, we conclude that in no more than

$$\left\lceil \frac{\log\xi}{\log(1 - \eta\delta_{2s})} \right\rceil$$

steps, the projection error $\|\Delta(m)\|_2$ satisfies $\|\Delta(m)\|_2 \leq \xi$. Hence

$$\begin{aligned}W\left(\hat{z}^{(k,m)}\right) &\overset{(a)}{=} \left\|(I - Q^+Q)e^{(k,0)} - \Delta(m)e^{(k,0)}\right\|_2 \\ &\overset{(b)}{\leq} \left\|(I - Q^+Q)e^{(k,0)}\right\|_2 + \left\|\Delta(m)e^{(k,0)}\right\|_2 \\ &\overset{(c)}{\leq} \left(\left(1 - \delta_{2s}\lambda^{-1}_{\max}\{Q^TQ\}\right)^{\frac{1}{2}} + \|\Delta(m)\|_2\right)\left\|e^{(k,0)}\right\|_2 \\ &\leq \left(\left(1 - \delta_{2s}\lambda^{-1}_{\max}\{Q^TQ\}\right)^{\frac{1}{2}} + \xi\right) W\left(\hat{z}^{(k,0)}\right)\end{aligned}$$

where the equality: (a) follows by substituting (12) in (11); (b) follows from the triangular inequality; and (c) follows from Proposition 4.8 and the first assumption which guarantees that $\hat{z}_E^{(k,0)}$ is s-sparse and hence $z_E^* - \hat{z}_E^{(k,0)}$ is at most $2s$-sparse. ■

Remark 4.10: It follows from the previous analysis that we can replace the inner loop in Algorithm 1 with a one step projection of $\hat{z}$ on the kernel of Q, that is, Algorithm 1 can be simplified to the alternation between the following two steps:

$$\begin{aligned}\hat{z}^{(k+1)} &:= \hat{z}_\Pi^{(k)} + Q^+\left(Y - Q\hat{z}_\Pi^{(k)}\right) \\ \hat{z}_\Pi^{(k+1)} &:= \Pi\left(\hat{z}^{(k+1)}\right).\end{aligned}$$

However, since computing the pseudo inverse matrix Q^+ can, in general, suffer from numerical issues, we argue that using the gradient descent algorithm (or any other recursive implementation that computes Q^+, e.g., Newton method, conjugate gradients, ...) is preferable.

3) Termination of the ETPG's Inner Loop: In the following result, we establish sufficient conditions for termination of the inner loop.

Proposition 4.11: Let the linear control system defined by (1) and (2) be $2s$-sparse observable with $2s$-restricted eigenvalue δ_{2s}. If the following holds:

1) $0 < \eta < \lambda^{-1}_{\max}\{Q^TQ\}$;
2) $\delta_{2s} > (4/9)\lambda_{\max}\{Q^TQ\}$.

then, there exists constants $\beta \in]0,1]$ and $\nu \in \mathbb{R}^+$ such that after no more than

$$\left\lceil \frac{\log\left[\frac{1}{2}\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}\left(\beta+(1-\nu)^{\frac{1}{2}}-1\right)-\frac{1}{3}\beta\right]}{\log(1-\eta\delta_{2s})} \right\rceil \quad (13)$$

inner loop iterations m, the inner loop condition $V(\hat{z}_\Pi^{(k)}) \leq (1-\nu)V(\hat{z}_\Pi^{(k-1)})$ is satisfied.

Proof: Before we start, we recall that $V(\hat{z}_\Pi^{(k)}) = V(\hat{z}^{(k+1,0)}) = (1/2)\|Y - Q\hat{z}^{(k+1,0)}\|_2^2 = (1/2)\|Qe^{(k+1,0)}\|_2^2$. It follows from the definition of $e^{(k+1,0)} = z^* - \Pi(z^{(k,m)})$ that $e_E^{(k+1,0)}$ is at most $2s$-sparse. Accordingly, inequality (8) holds as follows:

$$\frac{\delta_{2s}}{2}W^2\left(\hat{z}_\Pi^{(k)}\right) \leq V\left(\hat{z}_\Pi^{(k)}\right) \leq \frac{\lambda_{\max}\{Q^TQ\}}{2}W^2\left(\hat{z}_\Pi^{(k)}\right). \quad (14)$$

Hence, if we can prove that applying $\lceil \log[(1/2)(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\})^{(1/2)}(\beta+(1-\nu)^{(1/2)}-1)-(1/3)\beta]/\log(1-\eta\delta_{2s})\rceil$ gradient descent steps followed by a projection step implies that:

$$W^2\left(\hat{z}_\Pi^{(k)}\right) \leq (1-\nu)\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}W^2\left(\hat{z}_\Pi^{(k-1)}\right) \quad (15)$$

holds, we can combine (15) with (14) to obtain

$$\begin{aligned} V\left(\hat{z}_\Pi^{(k)}\right) &\leq \frac{\lambda_{\max}\{Q^TQ\}}{2}W^2\left(\hat{z}_\Pi^{(k)}\right) \\ &\leq (1-\nu)\frac{\delta_{2s}}{2}W^2\left(\hat{z}_\Pi^{(k-1)}\right) \\ &\leq (1-\nu)V\left(\hat{z}_\Pi^{(k-1)}\right) \end{aligned}$$

and conclude that the inner loop condition is satisfied. Therefore, to finalize the proof we need to show that inequality (15) holds. This follows from

$$\begin{aligned} W\left(\hat{z}_\Pi^{(k)}\right) &= \left\|z^* - \Pi\left(\hat{z}^{(k,m)}\right)\right\|_2 \overset{(a)}{\leq} 2W\left(\hat{z}^{(k,m)}\right) \\ &\overset{(b)}{\leq} 2\left(\left(1-\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}+\xi\right)W\left(\hat{z}_\Pi^{(k-1)}\right) \end{aligned} \quad (16)$$

where the inequality (a) follows from Proposition 4.7 while inequality (b) follows from Proposition 4.9. It follows from the second assumption that:

$$\begin{aligned} &\frac{4}{9} < \delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\} \\ &\overset{(a)}{\Rightarrow} \frac{4}{9} \leq (1-\nu')^2\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\} \\ &\Rightarrow 2 \leq 3(1-\nu')\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}} \\ &\Rightarrow 2\left(1-\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}\right) \\ &\leq (1-\nu')\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}} \end{aligned} \quad (17)$$

where the implication (a) follows for $\nu' \in \mathbb{R}^+$ satisfying:

$$\nu' = \beta\left(1-\sqrt{\frac{4\lambda_{\max}\{Q^TQ\}}{9\delta_{2s}}}\right)$$

where $\beta \in]0,1]$ is a design parameter. Fix ν' and define ν'', ν''', and ξ such that $\nu' = \nu'' + \nu'''$ and $2\xi = (\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\})^{(1/2)}\nu'''$. It then follows from (17) that

$$2\left(1-\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}\right)+2\xi \leq (1-\nu'')\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}. \quad (18)$$

By combining the inequalities (16) and (18) we conclude that (15) holds by defining ν such that $(1-\nu) = (1-\nu'')^2$. The upper bound on the number of inner loop iterations follows from Proposition 4.9 by instantiating ξ as

$$\begin{aligned} \xi &= \frac{1}{2}\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}\nu''' \\ &= \frac{1}{2}\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}(\nu'-\nu'') \\ &= \frac{1}{2}\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}\left(\beta\left(1-\frac{2}{3}\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{-\frac{1}{2}}\right)-\nu''\right) \\ &= \frac{1}{2}\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}} \\ &\quad \times \left(\beta\left(1-\frac{2}{3}\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{-\frac{1}{2}}\right)+(1-\nu)^{\frac{1}{2}}-1\right) \\ &= \frac{1}{2}\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}\left(\beta+(1-\nu)^{\frac{1}{2}}-1\right)-\frac{1}{3}\beta \end{aligned}$$

and hence the number of inner loop iterations is bounded from above by

$$\begin{aligned} m &\leq \left\lceil \frac{\log\xi}{\log(1-\eta\delta_{2s})} \right\rceil \\ &= \left\lceil \frac{\log\left[\frac{1}{2}\left(\delta_{2s}\lambda_{\max}^{-1}\{Q^TQ\}\right)^{\frac{1}{2}}\left(\beta+(1-\nu)^{\frac{1}{2}}-1\right)-\frac{1}{3}\beta\right]}{\log(1-\eta\delta_{2s})} \right\rceil \end{aligned}$$

where the first inequality follows from Proposition 4.9 along with the definition of ξ which ensures that ξ satisfies $0 < \xi < 1$. ■

D. Convergence of ETPG Algorithm

Using the previous results, we can now show convergence of the ETPG algorithm.

Proof of Theorem 4.4: Convergence of the algorithm, follows directly from the termination of the inner loop shown in Proposition 4.11. That is, it follows from the inequality (14) that the sequence $\sqrt{(2/\delta_{2s})V(\hat{z}_\Pi^{(k)})}$ forms an upper bound on $W(\hat{z}_\Pi^{(k)})$. However, it follows from the theorem assumptions along with Proposition 4.11 that

$$V\left(\hat{z}_\Pi^{(k+1)}\right) \leq (1-\nu)V\left(\hat{z}_\Pi^{(k)}\right) \Rightarrow \lim_{k\to\infty}\sqrt{\frac{2}{\delta_{2s}}V\left(\hat{z}_\Pi^{(k)}\right)} = 0$$

from which we conclude

$$\lim_{k\to\infty} W\left(\hat{z}_\Pi^{(k)}\right) = 0$$

which in turn implies the existence of an outer loop index k such that the bound $\|z^* - \hat{z}_\Pi^{(k)}\|_2 \leq \varepsilon$ is satisfied. ■

V. An Event-Triggered Projected Luenberger Observer

In this section, we describe a solution to Problem 2.4 obtained by rendering the ETPG algorithm recursive.

A. The Algorithm

We start again with the linear dynamical system defined by (1) and (2). The dynamics of $z(t) = (x(t-\tau+1), E(t)) \in \mathbb{R}^n \times \mathbb{R}^{p\tau}$ can be written, using the equality $a(t) = y(t) - Cx(t)$, as follows. First, let b_i denote the ith natural basis vector and define the shift matrix S, G_i and H_i as:

$$S = \begin{bmatrix} 0 & 1 & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix}, \; G_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -C_i A^{\tau} \end{bmatrix}, \; N_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_i^T \end{bmatrix}$$

$$H_i = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & 0 \\ -C_i A^{\tau-1} B & -C_i A^{\tau-2} B & \dots & -C_i B \end{bmatrix}$$

and hence

$$\begin{bmatrix} x(t-\tau+1) \\ E_1(t) \\ \vdots \\ E_p(t) \end{bmatrix} = \begin{bmatrix} A & 0 & \dots & 0 \\ G_1 & S & \dots & 0 \\ \vdots & & \ddots & \\ G_p & 0 & \dots & S \end{bmatrix} \begin{bmatrix} x(t-\tau) \\ E_1(t-1) \\ \vdots \\ E_p(t-1) \end{bmatrix} + \begin{bmatrix} [B & 0 & \dots & 0] \\ & H_1 & & \\ & \vdots & & \\ & H_p & & \end{bmatrix} \begin{bmatrix} u(t-\tau) \\ u(t-\tau+1) \\ \vdots \\ u(t-1) \end{bmatrix} + \begin{bmatrix} 0 \\ N_1 \\ \vdots \\ N_p \end{bmatrix} y(t)$$

which can be written in compact form as

$$z(t) = \overline{A} z(t-1) + \overline{B} \bar{u}(t-1)$$

$$Y(t-1) = Q z(t-1)$$

where

$$\overline{A} = \left[\begin{array}{cccc} A & 0 & \dots & 0 \\ \hline G_1 & S & \dots & 0 \\ \vdots & & \ddots & \\ G_p & 0 & \dots & S \end{array}\right] \overline{B} = \left[\begin{array}{cccc|c} [B & 0 & \dots & 0] & 0 \\ \hline & H_1 & & & N_1 \\ & \vdots & & & \vdots \\ & H_p & & & N_p \end{array}\right]$$

$\bar{u}(t-1) = \begin{bmatrix} U(t-1) \\ y(t) \end{bmatrix}$, and $z(t), Y(t), U(t), Q$ are as defined as in Section II.

The Event-Triggered Projected Luenberger (ETPL) Observer consists of the iteration of the following two steps:

a) Time update: Starting from an estimate $\hat{z}(t-1)$, we use the dynamics to update the previous estimate

$$\hat{z}_T(t) = \overline{A} \hat{z}(t-1) + \overline{B} \overline{u}(t-1).$$

This may result in an increase in the value of the Lyapunov candidate function V.

b) Event-triggered projected luenberger (measurement) update: In this step, we iteratively apply the projection operator Π once

$$\hat{z}_\Pi(t) = \Pi\left(\hat{z}_T(t)\right)$$

followed by applying the Luenberger update step

$$\hat{z}^{(m+1)}(t) = \hat{z}^{(m)}(t) + L\left(Y(t) - Q\hat{z}^{(m)}(t)\right)$$

multiple times starting from $\hat{z}^{(0)}(t) = \hat{z}_\Pi(t)$. It follows from the proof of Theorem 4.4 that alternating between multiple Luenberger updates (which is the generalization of the gradient descent step when $L = Q^T\Sigma$ for some positive definite matrix Σ) and projection steps forces a decrease of the Lyapunov candidate function $V(\hat{z}(t)) = (1/2)\|Y(t) - Q\hat{z}(t)\|_2^2$. In order to compensate for the increase introduced by the time update step, we need to ensure that V decreases along the execution of the algorithm. Hence, multiple Luenberger updates and projection steps are performed for each time-update step. Using the same triggering technique, by monitoring the evolution of V, the algorithm can determine when the decrease in V caused by the Luenberger update/projection steps compensates the increase caused by the time-update. Finally, we assign the estimate $\hat{z}(t)$ after the Luenberger update loop terminates to be $\hat{z}(t) = \hat{z}^{(m)}(t)$. This sequence of steps results in Algorithm 2.

Algorithm 2 Event Triggered Projected Luenberger Observer

a) Time Update:
$\hat{z}_T(t) = \overline{A}\hat{z}(t-1) + \overline{B}\overline{u}(t-1)$;
b) Event-Triggered Projected Luenberger (Measurement) Update:
reset $\hat{z}_\Pi(t) := \Pi(\hat{z}_T(t))$, $\hat{z}(t) := \hat{z}_\Pi(t)$;
while $V(\hat{z}_\Pi(t)) \geq (1-\nu)V(\hat{z}_\Pi(t-1))$ **do**
  reset $m := 0$, $\hat{z}_\Pi(t) := \Pi(\hat{z}(t))$, $\hat{z}^{(0)}(t) := \hat{z}_\Pi(t)$;
  $V_{\text{temp}} := V(\hat{z}_\Pi(t))$;
  while $V_{\text{temp}} \geq (1-\nu)V(\hat{z}_\Pi(t))$ **do**
    $\hat{z}^{(m+1)}(t) := \hat{z}^{(m)}(t) + L(Y(t) - Q\hat{z}^{(m)}(t))$;
    $V_{\text{temp}} := V(\Pi(\hat{z}^{(m+1)}(t)))$;
    $m := m + 1$;
  end while
  $\hat{z}(t) := \hat{z}^{(m)}(t)$;
end while

To make the connection with the standard Luenberger observer clearer, assume that only one Luenberger/projection

update is required per time update. In this case, the ETPL observer can be written as

$$\begin{aligned}\hat{z}_T(t) &= \overline{A}\hat{z}(t-1) + \overline{B}\overline{u}(t-1)\\ &= \overline{A}\hat{z}^{(1)}(t-1) + \overline{B}\overline{u}(t-1)\\ &= \overline{A}\Big(\hat{z}^{(0)}(t-1) + L\Big(Y(t-1) - Q\hat{z}^{(0)}\Big)\Big) + \overline{B}\overline{u}(t-1)\\ &= \overline{A}\,\Pi\left(\hat{z}_T(t-1)\right) + \overline{B}\overline{u}(t-1)\\ &\quad + L'\left(Y(t-1) - Q\,\Pi\left(\hat{z}_T(t-1)\right)\right)\end{aligned}$$

which has the form of a standard Luenberger observer with gain $L' = \overline{A}L$ along with the projection Π step.

Remark 5.1: It follows from Remark 4.10 that we can replace the inner loop in Algorithm 2 with one update step if we fix the Luenberger gain L to be $L = Q^{+}$.

B. Convergence of the ETPL Observer

In this subsection, we discuss the convergence and performance of the ETPL observer. The main result of this subsection is stated in the following theorem.

Theorem 5.2 (Convergence of the ETPL Observer): Let the linear control system defined by (1) and (2) be $2s$-sparse observable system with $2s$-restricted eigenvalue δ_{2s}. If the following condition holds

$$\delta_{2s} > \frac{4}{9}\lambda_{\max}\{Q^T Q\}$$

then the dynamical system defined by the ETPL observer is a solution to Problem 2.4 whenever $L = Q^T\Sigma$ for any positive definite weighting matrix Σ satisfying $\lambda_{\max}\{\Sigma\} < \lambda^{-1}_{\max}\{Q^T Q\}$.

The proof of Theorem 5.2 is based on the Lyapunov candidate function

$$W\left(\hat{z}(t)\right) = \|z(t) - \hat{z}(t)\|_2 = \|e(t)\|_2$$

and follows the exact same argument used in the proof of Theorem 4.4 with the exception of the need to account for an increase in V caused by the time update step. Such increase is compensated by applying the Luenberger update loop multiple times. This increase is described in the next result.

C. Effect of Time Update

Proposition 5.3: Consider the linear control system defined by (1) and (2). The following inequality holds:

$$W\left(\hat{z}(t)\right) \leq \|\overline{A}\|_2 W\left(\hat{z}(t-1)\right)$$

whenever $\hat{z}(t) \in \mathbb{R}^n \times \mathbb{R}^{p\tau}$ and $\hat{z}(t-1) \in \mathbb{R}^n \times \mathbb{R}^{p\tau}$ are related by

$$\hat{z}(t) = \overline{A}\hat{z}(t-1) + \overline{B}\overline{u}(t-1).$$

Proof: The Lyapunov candidate function $W(\hat{z}(t))$ after applying the time update step, can be written as follows:

$$\begin{aligned}W\left(\hat{z}(t)\right) &= \|z^*(t) - \hat{z}(t)\|_2\\ &= \left\|\overline{A}z^*(t-1) + \overline{B}\overline{u}(t-1) - \overline{A}\hat{z}(t-1) - \overline{B}\overline{u}(t-1)\right\|_2\\ &= \left\|\overline{A}\left(z^*(t-1) - \hat{z}(t-1)\right)\right\|_2 \leq \|\overline{A}\|_2 W\left(\hat{z}(t-1)\right).\end{aligned}$$

■

Proof of Theorem 5.2: First, we note that the Luenberger update loop in Algorithm 2 is identical to the outer loop of Algorithm 1 with the loop guard $V(\hat{z}^{(k)}_{\Pi}) \leq \epsilon$ (Line 2 in Algorithm 1) replaced with $V(\hat{z}_{\Pi}(t)) \leq (1-\nu)V(\hat{z}_{\Pi}(t-1))$. Hence, it follows from Theorem 4.4 that "Event-Triggered Projected Luenberger Update" loop terminates. From the termination of the Luenberger update loop, we conclude that the increase caused by the time update (Proposition 5.3) can always be compensated. Accordingly, using the same argument that was used in the proof of Theorem 4.4, we conclude that $\lim_{t\to\infty} W(\hat{z}_{\Pi}(t)) = 0$ and hence the estimate converges to the desired value z^*. ■

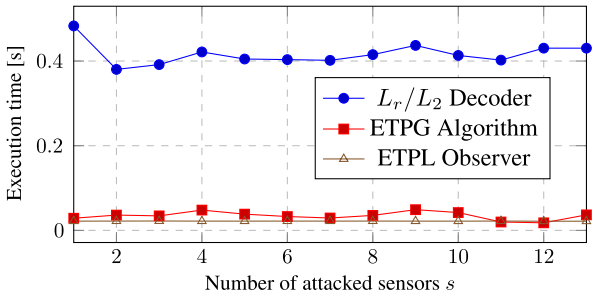
VI. Simulation Results

A. Computational Performance

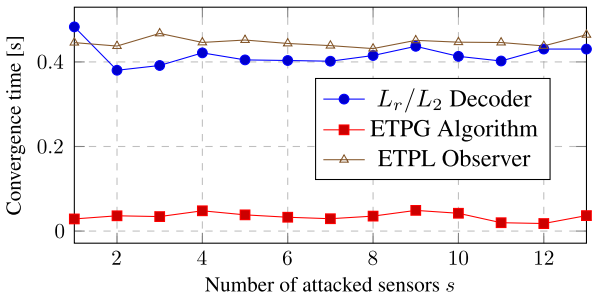
We compare the efficiency of the proposed ETPG and ETPL algorithms against the L_1/L_r decoder introduced in [10]. To perform this comparison, we randomly generated 100 linear systems with $n = 20$ and $p = 25$ and checked that the conditions of Theorems 4.4 and 5.2 hold. For each test case we simulated the linear systems against an increasing number of attacked sensors ranging from 0 to 12. In each test case we generated a random support set for the attack vector, a random attack signal, and random initial conditions. Averaged results for the different numbers of attacked sensors are shown in Fig. 2. Although we claim no statistical significance, the results in Fig. 2 are characteristic of the many simulations performed by the authors. The L_1/L_r decoder[1] is implemented using CVX while the ETPG and ETPL algorithms are direct implementations of Algorithms 1 and 2 in Matlab. The tests were performed on a desktop equipped with an Intel Core i7 processor operating at 3.4 GHz and 8 GB of memory.

Note that the ETPL observer, as required by any solution to Problem 2.4, is an asymptotic observer, i.e., the reconstructed state converges to the true state asymptotically. This should be contrasted with the ETPG algorithm where a single execution of Algorithm 1 is sufficient to construct an estimate which is ε-close to the system state. Hence, to compare the ETPG and ETPL algorithms we define the *execution time* as the time needed by Algorithms 1 and 2 to terminate and the *convergence time* as the time needed by each algorithm from the start of the execution until the estimate becomes ε-close to the system state. Note that the execution time affects the choice of the sampling period when the algorithm is deployed while the convergence time reflects the performance of each of the algorithms.

[1]We note that the L_1/L_r decoder is guaranteed to work only if the condition in Proposition 2 in [9] is satisfied. Since this condition requires applying a test for each point in the infinite set $\mathbb{R}^n$, it cannot be performed in practice. Hence, in these experiments, we run the L_1/L_r decoder and checked its success in reconstructing the state. In all our experiments, whenever conditions of Theorems 4.4 and 5.2 were satisfied, the L_1/L_r decoder was also successful in reconstructing the state.



(a) Execution time.



(b) Convergence time.

Fig. 2. Timing analysis of the L_1/L_2 decoder versus the ETPG and ETPL algorithms for different numbers of attacked sensors. (a) shows the time required to execute each of the three algorithms. (b) shows the time required for the estimate, computed by each of the algorithms, to become ϵ-close to the system state for $\epsilon = 10^{-6}$.

In Fig. 2(a), we can appreciate how both the ETPG and the ETPL algorithms outperform the L_1/L_r decoder[2] by an order of magnitude in execution time. We also observe that the ETPG algorithm requires more execution time compared to the ETPL observer. This follows from the existence of the outer-loop in the ETPG algorithm which requires the Lyapunov function to reach some small constant before termination. In Fig. 2(b), we show the convergence time for each of the three algorithms. It follows from the nature of the L_1/L_r decoder and the ETPG algorithm that the execution time and the convergence time are both equal. This is not the case for the ETPL observer which requires a longer convergence time compared to the ETPG algorithm. These two figures illustrate the tradeoff between execution timing and performance.

B. Multiple Attacking Sequences

In this example, we show how the proposed algorithms are still useful even in some cases where the conditions of Theorems 4.4 and 5.2 are not met. In particular, we consider an Unmanned Ground Vehicle (UGV) under different types of sensor attacks. We assume that the UGV moves along straight lines and completely stops before rotating. Under these assumptions, we can describe the dynamics of the UGV by

$$\begin{bmatrix}\dot{x}\\ \dot{v}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & \frac{-B}{M}\end{bmatrix}\begin{bmatrix}x\\ v\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{M}\end{bmatrix}F$$
$$\begin{bmatrix}\dot{\theta}\\ \dot{\omega}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & \frac{-B_r}{J}\end{bmatrix}\begin{bmatrix}\theta\\ \omega\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{J}\end{bmatrix}T$$

where x, v, θ, ω are the states of the UGV corresponding to position, linear velocity, angular position and angular velocity, respectively. The parameters M, J, B, B_r denote the mechanical mass, inertia, translational friction coefficient and the rotational friction coefficient. The inputs to the UGV are the force F and the torque T. The UGV is equipped with a GPS sensor which measures the UGV position, two motor encoders which measure the translational velocity and an inertial measurement unit which measures both rotational velocity and rotational position. We assume that encoders perform the necessary processing to directly provide a velocity measurement and hence the resulting output equation can be written as

$$y = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ v\\ \theta\\ \omega\end{bmatrix} + \begin{bmatrix}\psi_1\\ \psi_2\\ \psi_3\\ \psi_4\\ \psi_5\end{bmatrix}$$

where ψ_i is the measurement noise of the ith sensor which is assumed to be Gaussian with zero mean and finite covariance.

To obtain the discrete-time model of the UGV, we discretize the continuous-time model assuming a zero-order hold for the inputs F and T and using a time step equal to 0.01 s.

Although the framework in this paper considers attacks on all the sensors, all the results extend to a scenario where there are two groups of sensors: one that is prone to attacks (encoders) and one that is attack proof (GPS). Such change requires the $2s$-restricted eigenvalue to be computed, not by considering all possible sets Γ_{2s} of cardinality $2s$, but only those sets corresponding to encoder attacks. This new $2s$-restricted eigenvalue is nonzero and hence ETPG and ETPL algorithms can still be used to reconstruct the state whenever only one of the encoders are attacked. Although the framework in this paper considers attacks on all the sensors, all the results extend to a scenario where there are two groups of sensors: one that is prone to attacks (encoders) and one that is attack proof (GPS).

Fig. 3 shows the performance of the proposed algorithms under different attacks on the UGV motor encoders. The attacker alternates between corrupting the left and the right encoder measurements as shown in Fig. 3(c) and (d). Three different types of attacks are considered. First, the attacker corrupts the sensor signal with random noise. The next attack consists of a step function followed by a ramp. Finally a replay-attack is mounted by replaying the previously measured UGV velocity.

The UGV goal is to move 5 m along a straight line, stop and perform a 90° rotation and repeat this pattern three times until it traces a square and returns to its original position and orientation. In Fig. 3(a) and (b), we show the result of using the ETPG algorithm and the ETPL observer to reconstruct the state under sensor attacks. The reconstructed state is used by a linear feedback tracking controller forcing the UGV to track the desired square trajectory.

The reconstructed position and velocity are shown in Fig. 3(a) and (b). These figures show that both algorithms are able to successfully reconstruct the state and hence the UGV is able to reach its goal despite the attacks. Moreover, we observe that the ETPL observer is less sensitive to noise compared to the ETPG. This follows from the fact the ETPL observer "averages out" the noise by using all the available sensor data as it becomes available.

Recall that the attack model in Section II requires the set of attacked sensors to remain constant over time. However,

[2]In these experiments, we record the total CPU time as reported by the CVX log.

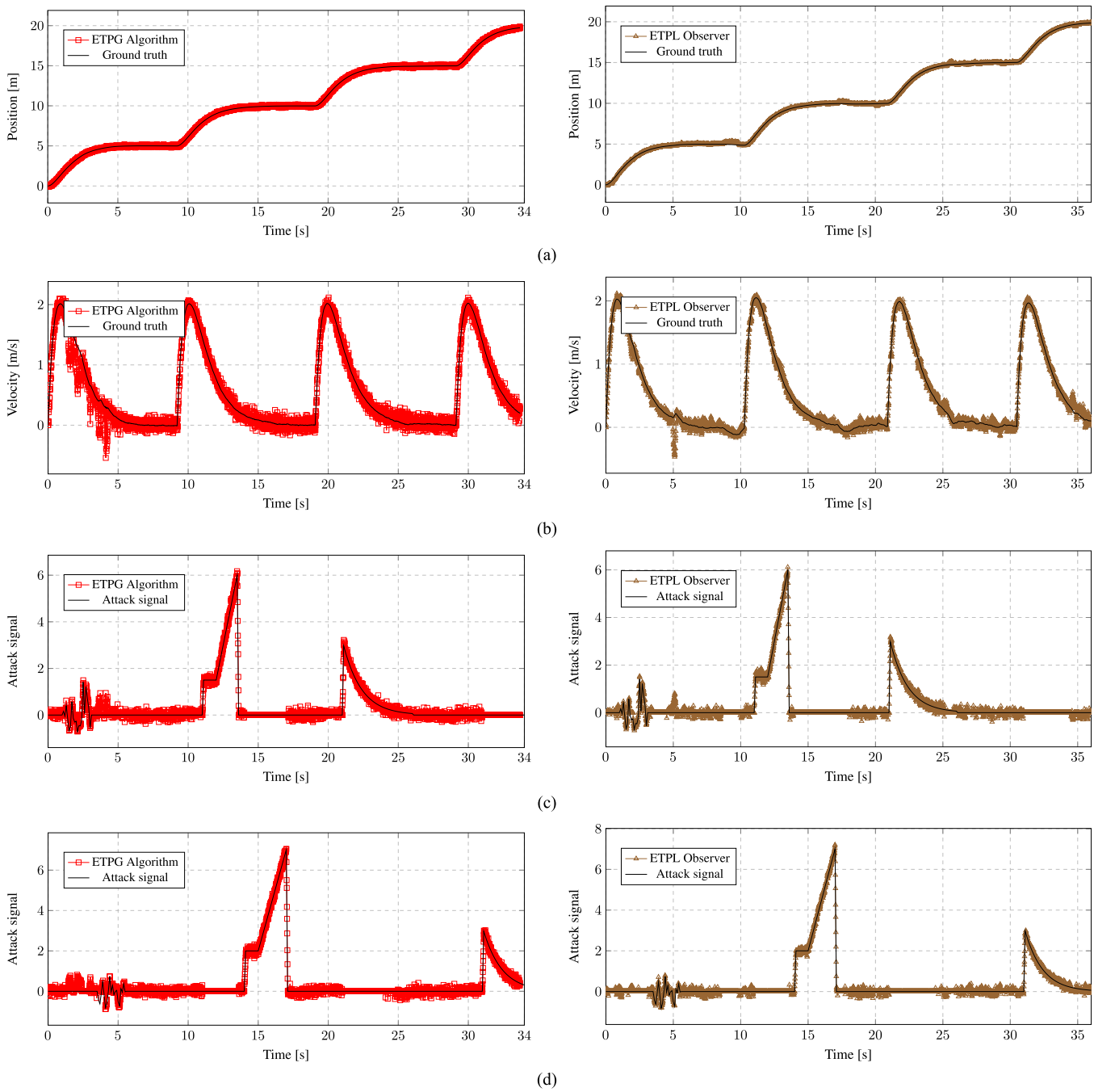


Fig. 3. Performance of the UGV controller in the cases where no attack takes place versus the case where the attack signal is applied to the UGV encoders. The objective is to move 5 m, stop and perform a 90° rotation and repeat this pattern to follow a square path. The controller uses ETPG algorithm and ETPL Observer to reconstruct the UGV states. In both cases we show the linear position (top), linear velocity (middle), and the reconstruction of the attack signal (bottom). (a) Reconstructed position versus ground truth. (b) Reconstructed velocity versus ground truth. (c) Reconstructed attack on left encoder versus ground truth. (d) Reconstructed attack on right encoder versus ground truth.

Fig. 3 shows the proposed algorithms correctly constructing the state even though this assumption is violated. This is due to the fact that the period during which only one sensor is attacked is sufficiently long compared with the time it takes for the algorithms to converge.

VII. Conclusion

In this paper, we considered the problem of designing computationally efficient algorithms for state reconstruction under sparse adversarial attacks/noise. We characterized the solvability of this problem by using the notion of sparse observability and proposed two algorithms for state reconstruction. To improve the timing performance of the proposed algorithms, we adopted an event-triggered approach that determines on-line how many gradient steps should be executed per projection on the set of constraints. These algorithms can be further improved along multiple directions such as dynamically adjusting the step size or using more refined gradient algorithms, such as conjugated gradient.

REFERENCES

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

[2] Y. Shoukry, P. D. Martin, P. Tabuada, and M. B. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *Workshop on Cryptographic Hardware and Embedded Systems*, G. Bertoni and J.-S. Coron, Eds., 2013, ser. CHES 2013, LNCS 8086, International Association for Cryptologic Research, pp. 55–72.

[3] K. Sou, H. Sandberg, and K. Johansson, "On the exact solution to a smart grid cyber-security analysis problem," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 856–865, 2013.

[4] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proc. 16th ACM Conf. Computer and Communications Security*, New York, 2009, ser. CCS'09, pp. 21–32, ACM.

[5] G. Dàn and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," in *Proc. 1st IEEE Int. Conf. Smart Grid Communications (SmartGridComm)*, 2010, pp. 214–219.

[6] O. Kosut, L. Jia, R. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[7] T. Kim and H. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 326–333, 2011.

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

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

[10] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure state-estimation for dynamical systems under active adversaries," in *Proc. 49th Annu. Allerton Conf. Communication, Control, Computing (Allerton)*, pp. 337–344.

[11] H. Fawzi, P. Tabuada, and S. Diggavi, "Security for control systems under sensor and actuator attacks," in *Proc. IEEE 5th Annu. Conf. Decision and Control (CDC)*, pp. 3412–3417.

[12] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[13] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok, "Introduction to compressed sensing," in *Compressed Sensing: Theory and Applications*, Y. C. Eldar and G. Kutyniok, Eds. Cambridge, U.K.: Cambridge University Press, 2011.

[14] J. Mattingley and S. Boyd, "Real-time convex optimization in signal processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 50–61, May 2010.

[15] M. S. Asif and J. K. Romberg, "Sparse recovery of streaming signals using l1-homotopy," *CoRR*, vol. abs/1306.3331, 2013.

[16] T. Blumensath, "Accelerated iterative hard thresholding," *Signal Process.*, vol. 92, no. 3, pp. 752–756, 2012.

[17] Y. Shoukry and P. Tabuada, "Event-triggered projected luenberger observer for linear systems under sensor attacks," in *Proc. IEEE 53rd Annu. Conf. Decision and Control (CDC)*, pp. 3548–3553.

[18] W. Kratz, "Characterization of strong observability and construction of an observer," *Linear Alg. and its Applic.*, vol. 221, no. 0, pp. 31–40, 1995.

[19] S. Sundaram and C. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1495–1508, Jul. 2011.

[20] G. Raskutti, M. J. Wainwright, and B. Yu, "Restricted eigenvalue properties for correlated gaussian designs," *J. Mach. Learn. Res.*, vol. 99, pp. 2241–2259, Aug. 2010.

[21] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[22] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[23] B. Amit and K. Eugenius, *Control Perspectives on Numerical Algorithms and Matrix Problems*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2006.

[24] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.

[25] T. Blumensath and M. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE J. Select. Topics Signal Process.*, vol. 4, no. 2, pp. 298–309, Apr. 2010.

[26] W. Petryshyn, "On generalized inverses and on the uniform convergence of $(I - \beta K)^n$ with application to iterative methods," *J. Mathemat. Anal. and Applic.*, vol. 18, no. 3, pp. 417–439, 1967.

Yasser Shoukry received the B.Sc. and M.Sc. degrees (with distinction and honors) from the Computer and Systems Department, Ain Shams University, Cairo, Egypt, in 2007 and 2010, respectively. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, University of California, Los Angeles.

He is affiliated with both the Cyber-Physical Systems Lab as well as the Networked and Embedded Systems Lab. In the summer of 2014 and spring of 2015, he was a visiting scholar at the University of California, Berkeley. Before joining the University of California at Los Angeles (UCLA), he was a Research and Development Engineer for four years, where he worked in the domain of automotive embedded systems and model-driven architecture. His research interests include the design and implementation of secure cyber-physical systems by drawing on tools from control theory, optimization theory, embedded systems, and formal methods.

Mr. Shoukry is the recipient of the Chancellors prize, the Graduate Division Fellowship, and the Preliminary Exam Fellowship, all from UCLA in 2011 and 2012.

Paulo Tabuada was born in Lisbon, Portugal, one year after the Carnation Revolution. He received the Licenciatura degree in aerospace engineering from the Instituto Superior Tecnico, Lisbon, Portugal, in 1998 and the Ph.D. degree in electrical and computer engineering in 2002 from the Institute for Systems and Robotics, a private research institute associated with the Instituto Superior Tecnico.

Between January 2002 and July 2003, he was a postdoctoral researcher at the University of Pennsylvania. After spending three years at the University of Notre Dame as an Assistant Professor, he joined the Electrical Engineering Department at the University of California at Los Angeles (UCLA), where he established and directs the Cyber-Physical Systems Laboratory. His latest book, Verification and Control of Hybrid Systems (New York: Springer, 2009).

Dr. Tabuada's contributions to cyber-physical systems have been recognized by multiple awards, including the National Science Foundation CAREER Award in 2005, the Donald P. Eckman Award in 2009, and the George S. Axelby Award in 2011. In 2009, he co-chaired the International Conference on Hybrid Systems: Computation and Control (HSCC'09) and in 2012, he was Program Co-Chair for the 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12). He also served on the Editorial Board of the IEEE Embedded Systems Letters and the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.