Self-Triggered and Event-Triggered Set-Valued Observers

Daniel Silvestre, Paulo Rosa, João P. Hespanha, Carlos Silvestre

Abstract

This paper addresses the problem of reducing the required network load and computational power for the implementation of Set-Valued Observers (SVOs) in Networked Control System (NCS). Event- and self-triggered strategies for NCS, modeled as discrete-time Linear Parameter-Varying (LPV) systems, are studied by showing how the triggering condition can be selected. The methodology provided can be applied to determine when it is required to perform a full ("classical") computation of the SVOs, while providing low-complexity state overbounds for the remaining time, at the expenses of temporarily reducing the estimation accuracy. As part of the procedure, an algorithm is provided to compute a suitable centrally symmetric polytope that allows to find hyper-parallelepiped and ellipsoidal overbounds to the exact set-valued state estimates calculated by the SVOs. By construction, the proposed triggering techniques do not influence the convergence of the SVOs, as at some subsequent time instants, set-valued estimates are computed using the *conventional* SVOs. Results are provided for the triggering frequency of the self-triggered strategy and two interesting cases: distributed systems when the dynamics of all nodes are equal up to a reordering of the matrix; and when the probability distribution of the parameters influencing the dynamics is known. The performance of the proposed algorithm is demonstrated in simulation by using a time-sensitive example.

Index Terms

State estimation; Fault detection; Self-triggered; Networked Control Systems.

I. Introduction

In the context of distributed systems and Networked Control Systems (NCSs), the performance bottleneck is often in the communication network, either due to low bandwidth, competition for access to a shared medium of communication, or because the network is much slower than the remaining components of the control loop. In distributed systems, different nodes are typically running an algorithm to achieve a certain goal and are only allowed to use information from their communicating neighbors. In networked control systems, sensors might be spatially spread over a region of interest and, therefore, measurements have to be sent to a controller/observer over the network. In any of such cases, the network resources are valuable and the communication issues must be considered as they can prevent the stability as given in [1] and [2]. For further details on this topic, the reader is referred to the detailed survey in [3], [4], [5] and [6]; and the book [7].

In the control community, two main strategies have emerged to reduce the number of communications, namely: event triggering, where the sensor decides, based on the current measurements, if it should transmit to the controller/observer the measured quantities; self triggering, where the controller/observer decides, based on the current estimate of the state, when the sensor should perform the next measurement. An event-triggered solution results in a more informed choice, since the sensor has access to the actual measurement, but prevents the sensor from being shut down between updates. For a recent discussion on event- and self-triggered control and estimation, the reader is referred to [8].

The problem of state estimation for general discrete-time Linear Parameter-Varying (LPV) systems relates to that of determining the possible future values that the state can take for a given set of inputs, initial state, measurements, and (deterministic) bounds on the noise and disturbances affecting the system. Two interesting instances of the state estimation problem can be found in the following contexts:

- asynchronous distributed algorithms - determining the state of each of the nodes given partial measurements and knowledge of the whole system dynamics;
- networked control systems - the observer must generate an estimate of the state and decide when to require a sensor update or define event conditions for the sensors to take that decision.

D. Silvestre is with the Department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Macau, Macau, China, and with the Institute for Systems and Robotics (ISR), Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal. D. Silvestre was supported by the project MYRG2016- 00097-FST from the University of Macau, by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through Institute for Systems and Robotics (ISR), under Laboratory for Robotics and Engineering Systems (LARSyS) project UID/EEA/50009/2013 and with grant SFRH//BD/71206/2010, from FCT. dsilvestre@umac.mo

P. Rosa is with Deimos Engenharia, Lisbon, Portugal. paulo.rosa@deimos.com.pt.

C. Silvestre is with the Department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Macau, Macau, China, on leave from Instituto Superior Técnico/Technical University of Lisbon, 1049-001 Lisbon, Portugal. The work was supported by project MYRG2016-00097-FST of the University of Macau. csilvestre@umac.mo

João P. Hespanha is with the Dept. of Electrical and Computer Eng., University of California, Santa Barbara, CA 93106-9560, USA. J. Hespanha was supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under grants No. W911NF-09-1-0553 and W911NF-09-D-0001. hespanha@ece.ucsb.edu

The design choice of considering LPV systems to model NCSs arises from the ability of these models to represent different (possibly unknown) dynamics. For instance a simple mechanical system, connected through a network to a controller, might have its dynamics dependant on the mass of some component and its value unavailable to the observer. Other example would be a sensor network where nodes are running a distributed algorithm. The overall dynamics depends on which nodes are transmitting information. The ability of LPVs to deal with uncertainty and have the remaining system matrices depend on some measurable parameters motivated its use to model NCSs.

Within the aforementioned framework, this paper is concerned with obtaining set-valued estimates of the state of the system that are guaranteed to contain the true state. The approach of using Set-Valued Observers (SVOs) first introduced in [9] and [10], is adopted - further information can be found in [11], [12] and the references therein. The SVO paradigm has the advantage of posing mild assumptions on the system, while allowing for the computation of *a priori* bounds for the maximum error. However, the computational cost is still one of the main issues associated with using SVOs (see [13]). In the remainder of this paper, this limitation is tackled by resorting to event- and self-triggered control systems techniques.

The adoption of a mathematical formulation for representing the set of possible states entails the need for fast and non-conservative intersections and unions of sets, as those are the major time-consuming operations when implemented in a computer. An alternative would be to use the concept of zonotopes, described in [14] and further developed in [15] and [16]. However, these represent a different compromise between the speed of the unions and intersections, with the intersections requiring more computations and introducing conservatism. Alternatively, the idea of interval analysis [17] may also be adopted, although it introduces conservatism by not considering higher horizon values in their formulation, unlike the SVOs [18]. Any set-based approach differs from other traditional methods, employing for example H_∞ filters [19], in the sense that it is obtained all possible instantiations of the system state, which is ideal for event-triggered strategies without the need for defining threshold values.

Throughout this paper, we focus on two main applications to motivate the theoretical developments of the SVO: the use of SVOs to obtain set-valued state estimates of event- and self-triggered networked control systems; and, ~~the example of~~ fault detection in randomized distributed systems. Both are intrinsically related in the sense that, in both cases, the goal is ~~set~~ to minimize either the sensor updates or the computational burden associated with the set-valued estimates computations to reduce the cost of implementation of this method in such systems.

The strategy for an observer to self-trigger a sensor measurement based on its estimates can resort to an optimization over the update patterns such as in [20], where the disturbances and noise are assumed to be Gaussian. In [21] and [22], Kalman-like filters are proposed for state estimation, thus not providing a deterministic bound for the error. For event-triggered systems, a condition can be posed on the norm of the estimation error being below a given threshold, dependent on the norm of the state [23]; requiring the derivative of a Lyapunov function of the state being semi-negative definite [8], [24]; or, having the norm of the state below a certain threshold [25].

The aforementioned methods differ from the current proposal in the following key aspects: by defining the triggers based on the stochastic distribution of the state (i.e., when involving a Kalman filters); by performing a complex optimization over update patterns; or by establishing the criteria on some type of norm exceeding a threshold. In this paper, event- and self-trigger strategies are investigated for networked control systems with the objective of developing an online strategy based on set-valued estimates, which means that, at each time instant, the observer produces a polytope, where the state is guaranteed to belong, and either triggers or allows the sensor to decide the next time instant to perform a measurement update.

An important topic to be discussed concerns the complexity and network load issues of these strategies. A Kalman filter solution, although of light complexity, fails to provide worst-case guarantees as the decision relies on the probabilistic distribution of the state. In addition, designing event-trigger strategies is troublesome since triggering is based on a threshold imposed to the variance and not on the particular measurements. The remaining strategies revolve around the concept of measuring the energy of the state in some way. These are connected to an SVO-based approach in the sense that both define sets of admissible state values and otherwise a trigger is generated. There is an inherent trade-off between accuracy and complexity. In particular, for LPV systems, a better accuracy provided by the SVOs represents a higher computational cost but might enable a triggering strategy that demands fewer sensor updates.

Fault Detection and Isolation (FDI) has been a long-standing research topic, since the early 70's (see [26]), but still poses remarkable challenges to both the scientific community and the industry (see, for example, the survey in [27] and references therein). Classical fault detection methods such as the ones proposed in [26], [28], [29] and [30], rely on designing filters that generate residuals that should be *large* under faulty environments. These strategies aim to derive bounds (or thresholds) on these residuals that can be used to decide whether a fault has occurred or not. However, the calculation of these thresholds is typically cumbersome or poses stringent assumptions on the exogenous disturbances and measurement noise acting upon the system. In contrast, SVOs aim to compute a set-valued estimate of the state with mild assumptions such as the existence of an overbound for all the signals in the system.

In the context of fault detection, focus is given to the special case of randomized distributed algorithms (see [31] for details on fault detection in this class of systems), for two reasons: their relevance in certain problems — applications range from selection and sorting [32] to consensus [33] and solving high-complexity problems; and, because of their unstructured nature, i.e., all nodes play the same role in the algorithm, while the messages need not satisfy any particular type of time sequence,

since any two messages are regarded as having the same purpose. Detecting faults in a distributed way in this setup may represent a constant huge overhead in terms of computation, while a self- or event-triggered strategy may yield similar results, although not running the procedure to obtain the set-valued estimates at every instant and choosing only the updates that contribute to the detection.

The class of problems addressed poses challenges to the state estimation scheme since, due to the random behavior of gossip algorithms or the network medium, for each possible sensor transmission, the state can belong to a set of possible state realizations originated by the dynamics and the previous state. To consider the worst-case scenario, one needs to perform the union of all possible state sets, which, in general, returns a non-convex set [34]. Furthermore, the number of sets grows exponentially with the number of past time instants considered, i.e., the horizon N. As a result, appropriate tools must be employed to reduce this complexity.

The main contributions of this paper can be summarized as follows:

- Given a specific structure for the matrix defining the polytope (i.e., the set-valued state estimate), it is shown how to compute an overbounding hyper-parallelipiped, ellipsoid, or ball;
- Based on the concept of singular vectors, we show how a rotation can be found to prevent the approximation error of using boxes from going to infinity when the matrix defining the polytope is ill-conditioned;
- For the special case of a distributed linear algorithm with a gossip property, it is shown that the overbounds are efficient to compute and propagate, since its complexity is constant;
- It is described how the set-valued state estimates provided by the SVOs can be used to define event- and self-triggering conditions for NCS;
- An algorithm is introduced that uses overbounding methods to approximate the *optimal* SVO estimates, which is less computationally demanding, and event- and self-triggers the computation of the aforementioned estimates only when necessary to ensure convergence;
- Results are provided regarding the worst-case frequency of the triggers for a class of LPV systems and its counterpart for randomized algorithms;
- Finally, it is given an improved result for convergence that takes into consideration the structure of the ouput equation of the LPV system.

The remainder of this paper is organized as follows. In Section II, it is described the problem of state estimation using SVOs and highlighted the main challenges. In Section III, it is shown how to find hyper-parallelipiped approximations and reduce its conservativeness, as well as how to compute ellipsoidal approximations. The use of SVOs to define Event- and Self-Triggered NCS is detailed in Section IV, whereas Section V presents Event- and Self-Triggered SVOs. Convergence results and regarding the triggering frequency are provided in Section VI and illustrated in simulation in Section VII. Concluding remarks are provided in Section VIII.

Notation : The transpose of a matrix A is denoted by $A^\intercal$. For vectors a_i, $(a_1, \dots, a_n) := [a_1^\intercal \ \dots a_n^\intercal]^\intercal$. We let $1_n := [1 \dots 1]^\intercal$ and $0_n := [0 \dots 0]^\intercal$ indicate n-dimensional vectors of ones and zeros, respectively, and I_n denotes the identity matrix of dimension n. Dimensions are omitted when clear from context. The vector e_i denotes the canonical vector whose components are equal to zero, except for the ith component which is equal to one. The symbol $\otimes$ denotes the Kronecker product. The notation $||.||$ refers to $\|v\| := \sqrt{v^\intercal v}$ for a vector, and $\|A\| := \sigma_{\max}(A)$ for a matrix, where $\sigma_{\max}(A)$ is the maximum singular value of matrix A. The orthogonal group of dimension n is represented by $\mathrm{O(n)}$.

II. Problem Statement

In this paper, we are addressing the problem of estimating the set of possible state values for a distributed system or a networked control system, which can be described by a discrete-time Linear Parameter-Varying (LPV) system of the form:

$$\begin{cases} x(k+1) = \Big(A_0 + \sum_{\ell=1}^{n_\Delta} \Delta_\ell(k) A_\ell\Big) x(k) + B(k)u(k) + L(k)d(k) \\ y(k) = C(k)x(k) + \nu(k) \end{cases} \tag{1}$$

with bounded unknown exogenous disturbances, $d(k) \in \mathbb{R}^{n_d}$, bounded unknown sensor noise, $\nu(\cdot)$, and uncertain initial state $x(0) \in X(0)$, where $X(0)$ is a known polytope. It is assumed that $x(k) \in \mathbb{R}^{n_x}$ and the known exogenous input vector $u(k) \in \mathbb{R}^m$. Without loss of generality $|d_i(k)| \leq 1, \forall k \geq 0$ and $|\nu_i(k)| \leq \nu^\star$. The dynamics matrix, $A(k) := \Big(A_0 + \sum_{\ell=1}^{n_\Delta} \Delta_\ell(k) A_\ell\Big)$, is affine on the polytopic uncertainties $\Delta(k)$. In this paper, two different scenarios will be considered: the first one assumes no information about $\Delta(k)$; the second one considers that each $\Delta(k)$ is an independent and identically distributed process for which the probability distribution is known. When not mentioned, we are dealing with the broader case of no information. The value of n_Δ is the number of uncertainties and each $\Delta_\ell(k)$ is a scalar uncertainty with $|\Delta_\ell(k)| \leq 1$. We assume for the sake of simplicity that matrices $B(k)$, $L(k)$, and $C(k)$ are parameter-independent and known, which can be relaxed by employing the techniques described in [18]. The two following problems are addressed in this paper.

Problem 1 (Triggering in the worst-case): Use the Set-Valued Observers (SVOs) framework to specify event- and self-triggered measurements when parameter $\Delta(\cdot)$ has an unknown distribution.

Problem 2 (Triggering with stochastic information): Use the Stochastic Set-Valued Observers (SSVOs) framework to specify event- and self-triggered measurements when the probability distribution p_ℓ for each matrix A_ℓ is known.

A distributed system or a networked control system conform with the description given by (1), where matrix $A(k)$ represents the dynamics which depends on the transmission that might not be accessible to the observer, thus leading to parameter $\Delta(k)$ being unknown. Matrix $C(k)$ is either going to determine the sensors that are making a measurement update or be zero when in between updates.

To address Problem 1, construct a set including all possible state realizations and obtain a bound on the error (i.e., the size of the computed polytope), the SVOs framework from [18] is adopted, defining at transmission time k,

$$X(k) := \text{Set}(M(k), m(k))$$

as the smallest set containing all possible state realizations at time instant k, where $\text{Set}(M, m) := \{q : Mq \leq m\}$ represents a convex polytope, with the operator $\leq$ being understood component-wise.

More precisely, the initial state is assumed to satisfy $x(0) \in X(0)$ for a given polytope $X(0) := \text{Set}(M_0, m_0)$. If the parameter $\Delta(k)$ can be measured, one can find $\Delta^\star$ such that (1) is a model of the system without any unknown parameters. For that particular $\Delta^\star$, let us define the set $X(k+1) := \text{Set}(M_{\Delta^\star}(k+1), m_{\Delta^\star}(k+1))$, which contains all the possible states of the system at time $k+1$. For the known $\Delta^\star$, $X(k+1)$ can be described by the set of points, $\mathbf{x}$, satisfying

$$\underbrace{\begin{bmatrix} M(k)A_k^{-1} & -M(k)A_k^{-1}L_k \\ \bar{C}_{k+1} & 0 \\ 0 & \bar{I} \end{bmatrix}}_{M(k+1)} \begin{bmatrix} \mathbf{x} \\ d(k) \end{bmatrix} \leq \underbrace{\begin{bmatrix} m(k) + \tilde{u}(k,1) \\ \bar{y}(k+1) + \nu^\star 1 \\ 1 \end{bmatrix}}_{m(k+1)} \tag{2}$$

where $\bar{Z} := \begin{bmatrix} Z \\ -Z \end{bmatrix}$, for a matrix or vector Z, $\tilde{u}(k,N) := \sum_{\tau=1}^{N} M(k)(A_n^\tau)^{-1}B(k)u(k-\tau+1)$ and

$$A_k = A_0 + \sum_{\ell=1}^{n_\Delta} \Delta_\ell^\star A_\ell$$

This procedure assumes an invertible transition matrix. When this is not the case, the strategy in [35] can be adopted, solving the inequality

$$\begin{bmatrix} \bar{I} & -\bar{A}_k & -\bar{L}_k \\ 0 & 0 & \bar{I} \\ \bar{C}_{k+1} & 0 & 0 \\ 0 & M(k) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^- \\ \mathbf{d} \end{bmatrix} \leq \begin{bmatrix} \bar{B}_k u(k) \\ 1 \\ \bar{y}(k+1) - \nu^\star 1 \\ m(k) \end{bmatrix} \tag{3}$$

by applying the Fourier-Motzkin elimination method [36] to remove the dependence on $\mathbf{x}^-$ and obtain the set described by $M(k+1)\mathbf{x} \leq m(k+1)$.

In order to illustrate the SVO computations, assume an abstract system described by the Linear Time-Invariant (LTI) model:

$$\begin{cases} x(k+1) = \begin{bmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{bmatrix} x(k) + 0.1d(k) \\ y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k) + \nu(k) \end{cases} \tag{4}$$

where $\forall k \geq 0 : |\nu(k)| \leq 0.1$ and an initial state uncertainty $\forall i \in \{1,2\} : |x_i(0)| \leq 1$. The system has invertible dynamics and the set $X(1) = \text{Set}(M(1), m(1))$ given by

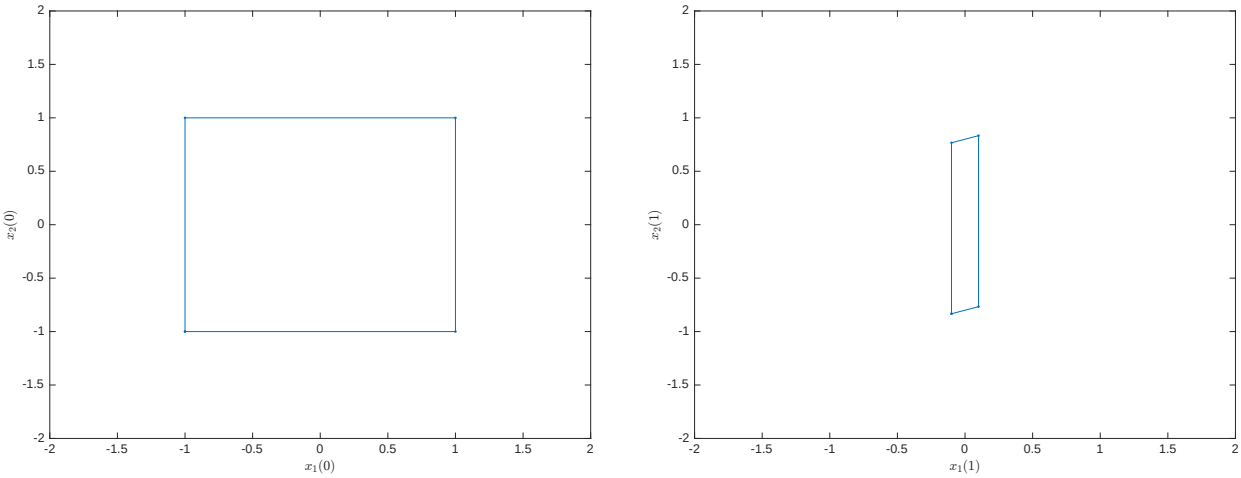
$$M(1) = \begin{bmatrix} 1.5 & -0.5 & -0.15 & 0.05 \\ -1.5 & 0.5 & 0.15 & -0.05 \\ -0.5 & 1.5 & 0.05 & -0.15 \\ 0.5 & -1.5 & -0.05 & 0.15 \\ 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, m(1) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

for the measurement $y(1) = 0$. The set $X(1)$ is exact as we have assumed an LTI with no uncertainty in its dynamics and depends on the variables $\begin{bmatrix} \mathbf{x}^\intercal & \mathbf{d}^\intercal \end{bmatrix}^\intercal$. The set $X(1)$ can be described solely by the variable $\mathbf{x}$ performing an elimination of

the $\mathbf{d}$ variable, obtaining

$$M(1) = \begin{bmatrix} 5 & -15 \\ -5 & 15 \\ 1 & 0 \\ -1 & 0 \end{bmatrix}, m(1) = \begin{bmatrix} -12 \\ -12 \\ 0.1 \\ 0.1 \end{bmatrix}$$

which we have depicted in Fig. 1. In the case of an LTI, the methods described in the remainder of this section are not required since the exact set $X(k)$ can be obtained.



(a) Initial uncertainty $X(0)$.

(b) Polytope $X(1)$.

Fig. 1. Example of the evolution of Algorithm 1 for a polytope that is not centered and not centrally symmetric. Edges are counted starting at the top one and counterclockwise.

Based on (2) and (3), a polytopic approximation $\tilde{X}(k+1)$ is computed by performing the convex hull of the union over all possible values of $\Delta(k+1)$. This is computationally more efficient than working with the non-convex set $X(k+1)$. Let the coordinates of each vertex of the hypercube $\mathcal{H} := \{\delta \in \mathbb{R}^{Nn_\Delta} : |\delta| \leq 1\}$ be denoted by $\theta_i, i = 1, \cdots, 2^{Nn_\Delta}$ for a general horizon N. Using (3), we compute $X_{\theta_i}(k)$ with $\Delta^\star = \theta_i$. Thus, the set of all possible states at time $k+1$ can be obtained as

$$\tilde{X}(k+1) = \text{co}\Big(\bigcup_{\theta_i \in \mathcal{H}} \text{Set}(M_{\theta_i}(k+1), m_{\theta_i}(k+1)) \Big) \tag{5}$$

where the union is done for all the vertices θ_i, and where M_{θ_i} and m_{θ_i} are obtained using (3). The convex hull function, co() of the set $X(k+1)$, $\tilde{X}(k+1)$, is then obtained by using the methods described in [37] since, in general, the set $X(k+1)$ is non-convex even if $X(k)$ is convex. For additional properties of the set $\tilde{X}(k)$, the interested reader is referred to [38] and the references therein.

In the context of Problem 2, the technique described in [38] corresponds to computing

$$\bar{X}(k+1) = \text{co}\Big(\bigcup_{\theta_i \in \Theta} \text{Set}(M_{\theta_i}(k+1), m_{\theta_i}(k+1)) \Big)$$

where Θ is a smaller collection of the vertices of $\mathcal{H}$ such that the probability of the state being contained in $\bar{X}(k+1)$ is greater than or equal to $1-\alpha$, and is therefore referred to as an α-confidence set.

These calculations can be extended to the case where $\tilde{X}(k)$ is computed not only based on $\tilde{X}(k-1)$, but also based on previous estimates, $\tilde{X}(k-2), \cdot, \tilde{X}(k-N)$, where N is the so-called horizon of the SVO. This method is based on standard *lifting* techniques and allows reducing the conservatism of the approach when $\tilde{X}(k-1)$ is larger than $X(k-1)$. For further details on this topic, the reader is referred to [18].

In the context of networked control systems, the problem being addressed can be summarized as how to use SVOs to determine event- and self-triggered strategies for when the sensors need to perform a measurement. The method should provide a current set-valued estimate for the state that can also characterize the maximum estimation error; defined as the greatest distance between the center and any point of the polytope.

For fault detection in a randomized distributed system, the objective is to reduce the amount of computations while ensuring that the set-valued estimates do not diverge. This is an important issue since the complexity grows exponentially both with the number of uncertainties (which depends on the number of possible transmissions) and also with the horizon N, since in (5), $\mathcal{H}$ is of size 2^{Nn_Δ}. This problem can render the SVOs inapplicable for some systems, especially those with stringent time constraints such as in real-time control applications.

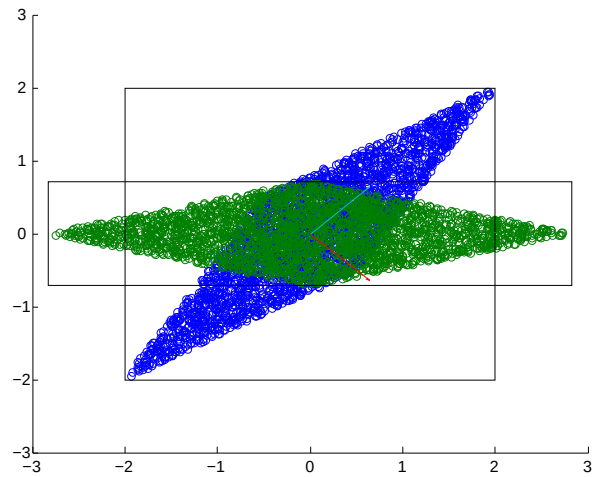


Fig. 2. Original and rotated sets, blue and green respectively, and its correspondent overbounds.

In the remainder of this paper, it will be discussed how to equip the SVOs with event- and self-triggered strategies to reduce the network resources requirements, and also reduce the complexity of the computations to a level where the state of time-sensitive applications can still be estimated resorting to an SVO-based technique.

III. Set-valued Estimate Approximations

A. Hyper-parallelepiped Approximation

The method to compute the set-valued state estimates described in the previous sections makes use of polytopes and produces approximations to the *optimal* SVO which are always of the form $\tilde{X}(k) = \{q : M(k)q \leq m(k)\}$. Without loss of generality, one can redefine these sets to be of the form $\tilde{X}(k) = \{q : M(k)q \leq 1\}$ assuming the origin is contained in the polytope. If this is not the case, one can simply shift the states so that the origin lies within the set.

The computation of unions and intersections of polytopes increases the number of vertices, which is a major limitation arising from the use of polytopes. A possible solution is to approximate $\tilde{X}(k)$ by a polytope with bounded number of vertices and obtain a set $\tilde{X}(k+1)$ that is more conservative than by simply considering, for instance, the convex hull. The additional error can be reduced by increasing the horizon N, as discussed previously. One possibility is to consider a hyper-parallelepiped overbound corresponding to the solution of the following linear program for each of the coordinate axis i

$$\begin{aligned} s_{2i-1} = &\underset{x}{\text{minimize}} \quad \mathsf{e}_i x \\ &\text{subject to} \quad M(k+1)x \leq 1, \end{aligned} \tag{6}$$

to get the minimum of this linear combination, while the corresponding maximum is obtained similarly by

$$\begin{aligned} s_{2i} = &\underset{x}{\text{minimize}} \quad -\mathsf{e}_i x \\ &\text{subject to} \quad M(k+1)x \leq 1, \end{aligned} \tag{7}$$

which generates the polytope $\tilde{X}(k+1) = \text{Set}(I \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix}, s)$ and can be put into the format where $m(k) = 1$ by dividing each of the rows of $I \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ by the corresponding entry in vector s. Notice that $m(k) > 0$ from the assumption that the origin is contained in the polytope, which can be obtained, in turn, by performing the required translation.

The hyper-parallelepiped approximation performance depends on the structure of the given polytope. As an example, consider the set provided in blue in Fig. 2, where the over-approximation would be a square of side 4. The area of the initial polytope is 8, whereas the area of its approximations is 16. If the set is "stretched" to get it closer to the line described by $y = x$, i.e., by increasing the condition number defined as

$$\kappa(M) = \frac{\sigma_{\max}(M)}{\sigma_{\min}(M)}$$

the overbound gets worse. In the following proposition, it is shown that the ratio between the hyper-volume of the set and the correspondent hyper-parallelepiped overbound can be arbitrarily large.

Proposition 1: Consider the hyper-parallelepiped approximation defined in (6) and (7) and a polytope $X = \text{Set}(M, 1)$.

$$\text{Then, } \exists M : \lim_{\kappa(M)\to\infty} \frac{\mathcal{V}(M')}{\mathcal{V}(M)} = \infty$$

where $\mathcal{V} : \mathbb{R}^{n\times m} \to \mathbb{R}$ maps a matrix $\bar{M}$ into the hyper-volume of $\text{Set}(\bar{M}, 1)$ and $X' = \text{Set}(M', 1)$ is the hyper-parallelepiped approximation of X computed using (6) and (7).

Proof. Take the polytope with matrix M given by

$$M = \left[\begin{array}{ccccc} -\frac{1}{\epsilon} & \frac{1}{\epsilon} & 0 & \cdots & 0 \\ \frac{1}{\epsilon} & -\frac{1}{\epsilon} & 0 & \cdots & 0 \\ \hline \multicolumn{5}{c}{I \otimes \bar{1}} \end{array}\right]$$

where solving (6) and (7) returns $M' = I \otimes \bar{1}$ for the hyper-parallelepiped X'. When $\epsilon \to 0$ we have $\kappa(M) \to \infty$ as $\sigma_{\max}(M) \to \infty$. We then have $\mathcal{V}(M') = 2^{n_x}$ while $\lim_{\kappa(M)\to\infty} \mathcal{V}(M) = 0$ which concludes the proof. ■

The key observation to improve the accuracy of the SVO calculation is to rotate the set in blue in Fig. 2 to align it with the coordinate axes (getting the set in green), obtaining in this way a less conservative overbound (in the example, the volumes goes from 16 to 8). The depicted vectors represent the singular vectors of matrix M and define the directions of principal components of the sets. Therefore, the relationship with the condition number is clear in the sense that the higher the condition number, the more one direction is less predominant when compared to the others. In the extreme case of a $\kappa(M) = \infty$, one can conclude that the set has zero length in one of the dimensions.

The solution proposed in this section to find an improved overbound is a rotation to get the singular vectors as the canonical basis, as demonstrated in Fig. 2. From the definition of the Singular Value Decomposition (SVD), $M = USV^\intercal$ where the right-singular vectors are the columns of V, which are orthonormal, and the singular values are the elements of the diagonal of matrix S. Matrix M can be seen as the set after the rotation of the canonical vectors to match its singular vectors, i.e., the original set is rotated with respect to the canonical basis as depicted in blue in Fig. 2. Then, to find the set with vectors aligned with the canonical basis depicted in green in Fig. 2, i.e., the set defined by M_{rot}, we can write the relationship between M and M_{rot} by the rotation matrix R as

$$M_{\text{rot}} = (RM^\intercal)^\intercal = MR^\intercal. \tag{8}$$

Matrix R can be obtained through the equation

$$RV = I \Leftrightarrow R = V^\intercal \tag{9}$$

as we want to rotate from the singular vectors in V to the canonical vectors. By combining (8) and (9) we get

$$\begin{aligned} M_{\text{rot}} &= (RM^\intercal)^\intercal \\ &= (V^\intercal(USV^\intercal)^\intercal)^\intercal \\ &= USV^\intercal V \\ &= US. \end{aligned}$$

Thus, a possible approach to reduce the conservatism of a hyper-parallelepiped approximation is to apply a rotation using the singular vectors. In doing so, the principal axes of the set are aligned with the canonical vectors and the resulting hyper-parallelepiped overbound is tighter.

We can now address the conservatism issue of the approximation for a general polytope and perform a similar analysis to Proposition 1 but after applying the rotation to the polytope. The next proposition shows that the ratio does not depend on the condition number and has a factor depending solely on the state dimension.

Proposition 2: Consider a polytope $X = \text{Set}(M, 1)$, where the singular vectors of M are the canonical vectors after applying the rotation defined in (9). Take the hyper-parallelepiped approximation $X' = \text{Set}(M', 1)$ of X, as defined in (6).

$$\text{Then, } \max_{M} \frac{\mathcal{V}(M')}{\mathcal{V}(M)} = n_x!$$

where $\mathcal{V}$ is defined as in Proposition 1.

Proof. We start by noticing that after the rotation, each of the hyper-faces of X' must contain at least a vertex of X which means that the worst case is to select the polytope X such that it has the lowest volume and at least a vertex in each of the hyper-faces of X'. This corresponds to select as X the n_x-simplex sharing n_x converging edges with the hyper-parallelepiped. In such case, we have

$$\mathcal{V}(M) = \frac{\mathcal{V}(M')}{n_x!}$$

which concludes the proof. ■

It should be noticed that for a general polytope X, the proposed rotation is not desirable in all cases. If we select a counterexample as in Fig. 3, the new set leads to a more conservative overbound. Nevertheless, the case in Fig. 3 is caused

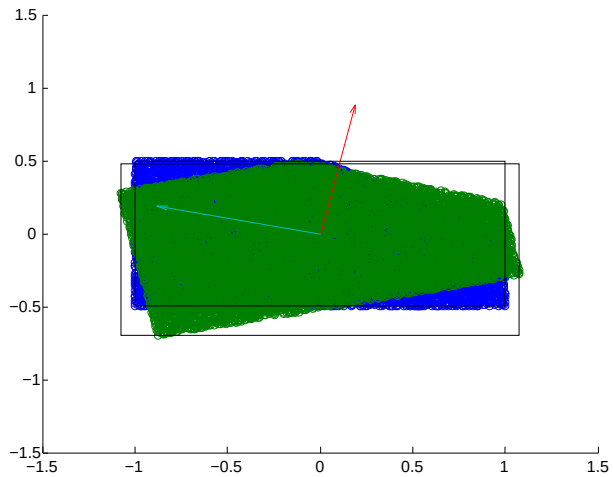


Fig. 3. Counterexample where a set is rotated but a worst overbound is achieved.

due to the lack of *central symmetry* of the polytope as in Definition 1. If the polytope is made centrally symmetric, then the proposed rotation ensures that the hyper-parallelepiped overbound has at most a factor of $n_x!$ increase in the hyper-volume of the set. Without using the rotation, the overbound can be arbitrarily large depending on the condition number of the matrix defining the polytope, as seen in Proposition 1.

Definition 1: A polytope $X :=\text{Set}(M, 1)$ is *centrally symmetric* if it can be written as the intersections of 2ℓ half-planes of the form, i.e. if M satisfies

$$M = \begin{bmatrix} m_1(k) \\ -m_1(k) \\ \vdots \\ m_\ell(k) \\ -m_\ell(k) \end{bmatrix}.$$

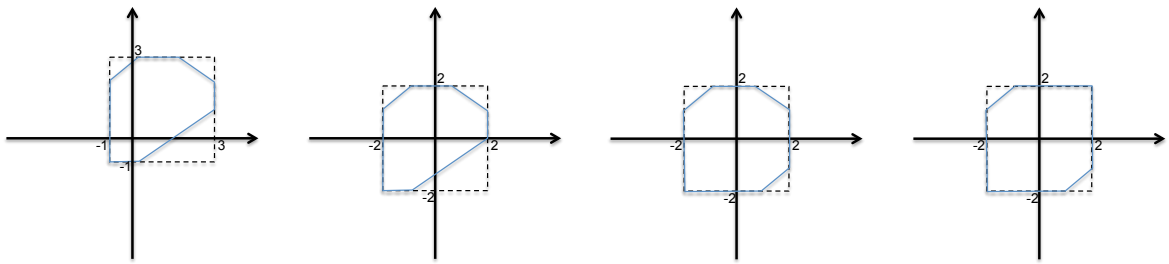
We now introduce an algorithm to make a polytope centrally symmetric, ensuring that the hyper-parallelepiped approximation after the rotation is not worse than the one before the rotation.

Algorithm 1 Centrally symmetric polytopes

Require: Polytope $X := \text{Set}(M, 1)$.
Ensure: Returns a polytope X which is centrally symmetric.

```
/* Center initial hyper-parallelepiped */
find s using (6) and (7)
apply translation to center hyper-parallelepiped defined by s
/* Add and remove rows */
for each row i do
    /* Test if intersects */
    if intersects(X, −m_i) then
        remove(X, m_i)
    else
        add(X, −m_i)
    end if
end for
return X
```

Algorithm 1 converts any general polytope in a centrally symmetric polytope with the important feature that the produced overbound does not increase the size of the hyper-polytopical and ellipsoidal techniques. The evolution of the set in different stages of the algorithm is illustrated in Fig. 4.



(a) Initial polytope and hyper-parallelepiped overbound. (b) Polytope after the translation. (c) Polytope after adding an edge symmetric to edge 2 and deleting edge 5. (d) Resulting, centrally symmetric polytope after removing edge 7.

Fig. 4. Example of the evolution of Algorithm 1 for a polytope that is not centered and not centrally symmetric. Edges are counted starting at the top one and counterclockwise.

B. Ellipsoidal Overbounding

The previous section introduced a rotation to deal with ill-conditioned cases in which a hyper-parallelepiped overbound degrades performance. By reducing the conservatism of the overbound, the SVO design can relax the need for a large horizon to ensure convergence. This section aims to overbound the set-valued estimates by an ellipsoid, with the ultimate objective of having an easy-to-compute estimate, in case the accuracy can be temporarily reduced in order to improve computational performance.

The main limitation of SVO-based techniques when applied to a real-time or time-sensitive application is the associated computational burden. In each iteration, generating the set containing all possible state realizations amounts to the union of the sets obtained by propagating all possible combinations of the system dynamics and intersecting it with the set of states compatible with the current measurements. This process may be time-consuming, especially when the model of the system is only partially known - see [38].

In the next theorem, it is shown how an ellipsoidal overbound can be computed for a generic polytope that satisfies the centrally symmetric condition of Definition 1.

Theorem 1: Consider a convex set

$$\mathcal{S} = \{x : Mx \leq 1\}$$

such that $M \in \mathbb{R}^{n\times m}$ is as in Definition 1. An ellipsoidal overbound to $\mathcal{S}$ is given by

$$x^\intercal Q x \leq 1,$$

where $Q = \frac{VS^\intercal SV^\intercal}{n}$.

Proof. The inequality $Mx \leq 1_n$ follows from the assumption that matrix M is as in Definition 1. We can infer $\forall x : Mx \leq 1_n \Rightarrow x^\intercal M^\intercal Mx \leq 1_n^\intercal 1_n$. After a singular value decomposition on M, the inequality becomes $\frac{1}{n}x^\intercal VS^\intercal U^\intercal USV^\intercal \leq 1$ and, since U is an orthogonal matrix, we get the conclusion. ■

A corollary of the previous result can be derived in order to provide an overbound in terms of the maximum norm of any point belonging to the set.

Corollary 1: Consider a convex set $\mathcal{S}$ such that matrix M is as in Definition 1. Then, an overbound to $\mathcal{S}$ can be described by

$$x^\intercal x \leq \frac{n}{\sigma^2_{\min}(M)}$$

which means that $||x|| \leq \frac{\sqrt{n}}{\sigma_{\min}(M)}$.

Proof. Given the result in Theorem 1, then $\forall x \in \mathcal{S} : x^\intercal Q x \leq 1$ with $Q = \frac{VS^\intercal SV^\intercal}{n}$. Then, $x^\intercal Q' x \leq x^\intercal Q x$ with $Q = \frac{\sigma^2_{\min}}{n} VIV^\intercal = \frac{\sigma^2_{\min}}{n} I$ which yields the result. ■

Remark 1: It should be stressed that alternative methods have been developed in the literature to obtain ellipsoidal set-valued state estimates, since the seminal work [10], as described, for instance, in [11]. However, the algorithm proposed in this work has some relevant properties, as discussed in the sequel, including the low-computational power required, as well as guaranteeing that the state of the system is indeed contained within the ellipsoid.

Recovering the abstract example in (4), the hyper-parallelepiped approximation would simply be the set described by the matrix M

$$M = \begin{bmatrix} 0 & \frac{6}{5} \\ 0 & \frac{-6}{5} \\ 10 & 0 \\ -10 & 0 \end{bmatrix}$$

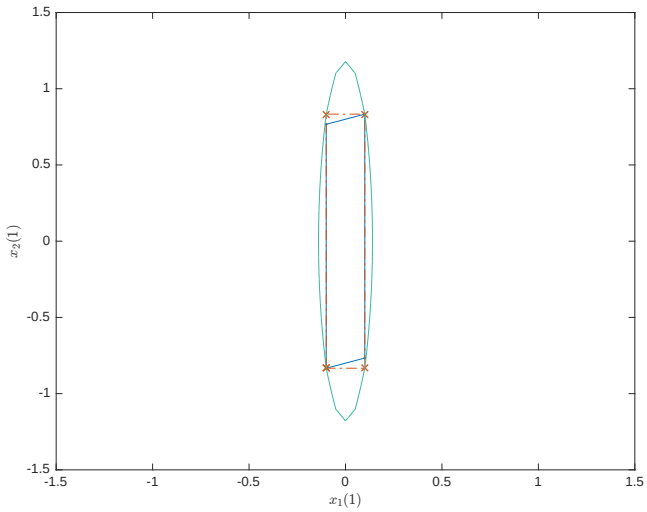


Fig. 5. Abstract example where the previous set $X(1)$ is enveloped by the hyper-parallelepiped approximation in dashed line and the ellipse upper bound.

and the ellipsoidal set would be given by matrix Q given by

$$Q = \begin{bmatrix} 50 & 0 \\ 0 & \frac{18}{25} \end{bmatrix}$$

which is depicted in Fig. 5 where for this abstract system the set $X(1)$ was not a particular bad choice as no rotation was needed.

IV. Set-Valued Observer for Event- and Self-Triggered Systems

The framework of SVOs deals with a worst-case scenario and provides set-valued estimates to which the state of the system is known to belong, in stark contrast with providing a single estimate and a probabilistic bound for the error for that estimate. In this section, we explore how to use the sets derived in the previous section to define conditions that, up to certain extent, generalize those surveyed in Section I with the clear benefit of enabling other shapes for the barrier condition for triggering a sensor measurement.

A. Set-Valued Observers for Event-Triggered Systems

Event-triggered systems aim to reduce the communication burden between the sensors and the observer, which in networked control systems makes use of the shared medium network, thus consuming resources that may be critical to the remaining processes using the network. In the literature, the event trigger condition is common to be defined at the expenses of the error of the last sensor update or as a quadratic or norm function of the state [25]. However, the set-valued estimates of the state can also be used to provide an *event condition* for the sensor to perform a measurement.

An SVO constructs the polytopic approximation set $\tilde{X}(k)$, at each time instant k, for a system described by (1). The objective is to use this information and find an event condition such that the sensor can determine when a measurement update is required. We introduce the notation $\tau^{-1}(k)$ to denote the last triggering time that is smaller than k. Similarly, $\tau^1(k)$ refers to the first occurrence of a trigger that is greater than k and $\tau^0(k) = k$ ($\tau^0(k)$ will be used instead of k whenever we want to state explicitly that the current k is a triggering time). The second most recent trigger is denoted by $\tau^{-2}(k) = \tau^{-1}(\tau^{-1}(k))$ and similarly for any other trigger.

A naive approach would entail the observer to send the matrix $M(\tau^{-1}(k))$ and the vector $m(\tau^{-1}(k))$ at time $\tau^{-1}(k)$ to the sensor, which assuming knowledge of the full state, then tests if

$$x(k) \in \tilde{X}(\tau^{-1}(k))$$

and, if the sensor has a partial observation, it can check the more general condition

$$M(\tau^{-1}(k))C(k)^{\dagger}y(k) \leq m(\tau^{-1}(k)) \tag{10}$$

where the symbol $\dagger$ stands for the Moore-Penrose pseudoinverse. For all subsequent time instants $\tau^{-1}(k) < k < \tau^1(k)$ the sensor needs to update matrix $M(\tau^{-1}(k))$, resorting to the nominal dynamics (the matrix A_0) and the control law $B(k)u(k)$. Condition (10) would easily not hold for cases where the stabilizing input signal has large magnitude. It is assumed that the

sensor receives the control signals from the controller as it is communicating to the plant. The update corresponds to computing $M(\tau^{-1}(k))A_0^{-1}$ and applying the translation given by $B(k)u(k)$. When (10) does not hold using the updated set, the sensor performs an update and the observer sends $\tilde{X}(\tau^0(k))$, which is the set for the current time.

The condition proposed in (10) can be viewed as a generalization of a condition depending on some norm. In particular, inequalities involving both the ℓ_∞ norm, defined as $\|x\|_\infty = \max_i |x_i|$, and the ℓ_1 norm, defined as $\|x\|_1 = \sum_{i=1}^n |x_i|$, are polytopes and can be represented in this framework. In addition, the observer/controller can place additional restrictions to trigger the update.

Example: Let us assume that the state of the system is a stock or other financial product quote and the observer/controller is a hedge fund manager running a control system to make the purchase and sell according to the received quotes. Due to regulation in the market, or motivated by correlation between products or even when having options and futures to cover the risk of other products, it might be useful to add new constraints to the transmitted condition. Such a condition cannot be represented using the previous norms. However, by using polytopes, extra linear restrictions can be represented by adding rows to $M(\tau^{-1}(k))$.

The conservatism of the initial set $X(0)$ (by assumption $\tilde{X}(0) := X(0)$) depends on the information available to the designer of the SVO. If little is known about the initial conditions of the systems, set $X(0)$ must be made sufficiently large so as to contain any possible initial state $x(0)$. Condition (10) would be meaningless in this case, as sensor readings would not be triggered. We introduce a performance parameter μ referring to the maximum allowed radius of the ball enclosing the polytope. Whenever

$$\frac{n_x}{\sigma_{\min}(M(k))} \geq \mu, \tag{11}$$

the observer requests a sensor measurement. The purpose of μ is to guarantee that the observer constructs a "reasonably good" estimate before setting an event condition.

The algorithm is summarized in Algorithm 2, where we use the notation $\neg$ as the logical negation.

Algorithm 2 SVOs for Event-Triggered systems

Require: Polytope $X(0)$.
Ensure: Event-triggered sensor updates.

```
for each k do
    if ¬(10) then
        sensor_update()
        X̃(τ^0(k)) =svo_update()
        if (11) then
            /* Force a trigger by sending an empty set instead of X̃(τ^0(k)) */
            send(empty_set)
        else
            send(X̃(τ^0(k)))
        end if
    end if
end for
```

In NCSs, where the use of the network is an extremely valuable resource, one can opt by reducing the communication of the event condition by applying any of the overbounding techniques described in this paper. If a hyper-parallelepiped or an ellipsoid approximation is used, the communication is reduced to the rotation matrix V of Theorem 1 multiplied by the expansion factors. Note that other techniques to reduce the size of transmitted information can be employed based on the exponential representation of the rotation matrix. If overbounding by a ball, the event condition resorts to an ℓ_2-norm and only the radius is required for the sensor to determine when to trigger a measurement.

B. Set-Valued Observers for Self-Triggered Systems

Self-triggered systems require the ability to propagate estimates into future time steps, so as to determine the next sensor reading. The SVOs have the capability of forward propagation to get the time instant for the next sensor measurement, provided that the volume of the set-valued estimates does not grow beyond a certain, predefined limit.

Inequality (3) defined the estimation set for the state in the next time instant using the knowledge of the sensor measurement $y(k)$. However, removing the rows corresponding to the intersection with the measurements, defines the set-valued estimates that results only from propagating the dynamics, which we denote by $\tilde{X}_p(k)$. At time $\tau^{-1}(k)$, the observer receives the measurement $y(\tau^{-1}(k))$, and has access to the set $\tilde{X}(\tau^{-2}(k))$. To determine the next sensor update, a node resorts to (3) to find $\tilde{X}(\tau^{-1}(k))$ and then propagates it using the dynamics in (1) to obtain $\tilde{X}_p(\tau^1(k))$, with $\tau^1(k)$ being the first time instant such that

$$\tilde{X}_p(\tau^1(k)) \nsubseteq \tilde{X}(\tau^{-2}(k)). \tag{12}$$

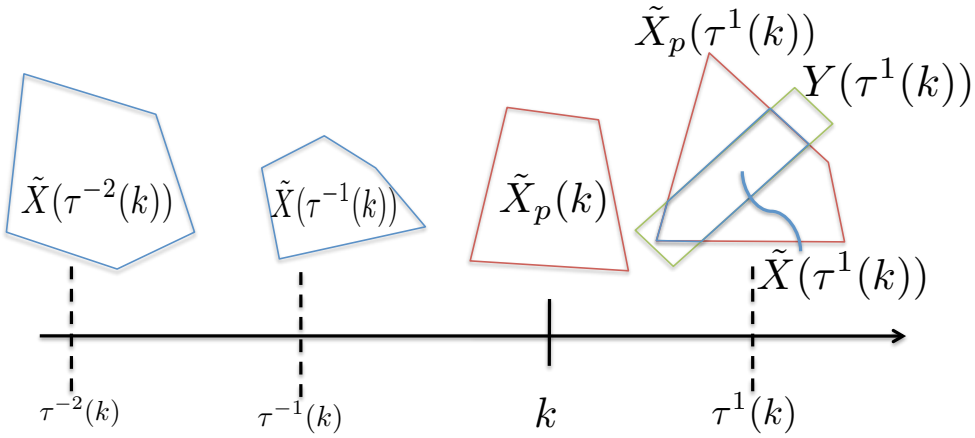


Fig. 6. Example of using SVOs for self-triggered systems. At time $\tau^{-1}(k)$, the observer computes set $\tilde{X}(\tau^{-1}(k))$ and propagates twice to get $\tilde{X}_p(k)$ and $\tilde{X}_p(\tau^1(k))$, which is larger than $\tilde{X}(\tau^{-2}(k))$, and triggers a sensor measurement, making the intersection with the measurement set $Y(\tau^1(k))$ to get the new estimation $\tilde{X}(\tau^1(k))$.

The condition assures that the observation set does not increase in size because of the self-triggered approach.

For the above approach, finding $\tau^1(k)$ can be performed by a logarithm search, testing different values for $\tau^1(k)$ as the size of $\tilde{X}_p(k)$ is monotonically increasing, unless we have the stringent condition that the singular values of any chain of dynamics matrix are smaller than 1. To account for the more general case, we select $\tilde{X}(\tau^{-2}(k))$ instead of $\tilde{X}(\tau^{-1}(k))$ in (12). The procedure is summarized in Algorithm 3.

Algorithm 3 SVOs for Self-Triggered systems

Require: Polytope $X(0)$.
Ensure: Self-triggered sensor updates.

for each $\tau^{-1}(k)$ **do**
  sensor_update()
  $\tilde{X}(\tau^{-1}(k))$ = *svo_update()*
  find $\tau^1(k)$ *such that* (12) *is satisfied*
  send($\tau^1(k)$)
end for

In systems where the computational power used in each time instant is limited, one can adopt a different strategy and have an iterative solution. By definition, we have the relationship $\tilde{X}(k) \subseteq \tilde{X}_p(k) \cap Y(k)$, since $\tilde{X}(k)$ is the convex hull of the intersection between $X_p(k)$ and $Y(k)$. To determine the set-valued estimate at time instant $\tau^{-1}(k)$, it is sufficient to compute $\tilde{X}_p(\tau^{-1}(k)) \cap Y(\tau^{-1}(k))$, which is an inexpensive computation. Instead of computing the set $\tilde{X}(\tau^1(k))$ for different values of future times, one can resort to pre-computed products of matrices using the values for the uncertainties. Then, proceed to check if each of the products exceeds $\tilde{X}(\tau^{-2}(k))$ in size, which is less computationally demanding since we prevented the computation of the convex hulls for all uncertainties for all values of next trigger time to be tested. The set $\tilde{X}_p(\tau^1(k))$ can be computed during the inactivity time between $\tau^{-1}(k)$ and $\tau^1(k)$, leaving $\tilde{X}(\tau^1(k))$ to be computed at time $\tau^1(k)$ from $\tilde{X}_p(\tau^1(k))$.

Figure 6 depicts how the sets in Algorithm 3 evolve with time. We draw attention to the fact that sets $\tilde{X}_p(k)$ are monotonically increasing in volume, which motivated the triggering condition to use the set at time instant $\tau^{-2}(k)$.

V. Event- and Self-Triggered Set-Valued Observers

In the prequel, SVOs were used to determine only the triggering of sensors update (i.e., when sensors are required to send measurements to the observer) in the context of NCSs where only the number of updates is minimized in a greedy approach. Nevertheless, the SVOs estimates are computed at every time instant, which motivates the introduction of Event- and Self-Triggered SVOs. The main objective is the reduction of the computational cost associated with the *classical* SVOs by only computing the set-valued state estimate (using the previously described tools) when the set is growing in size. As an alternative, we compute and propagate overbounds, which are less computationally demanding, and perform the intersection with the measurement set for the worst-case in terms of system dynamics. The methods described for reducing the complexity of the SVO computations are compatible with the previous use of SVOs for event- and self-triggered systems.

The main advantage of this method is its real-time application due to diminished computational costs associated with three main factors:

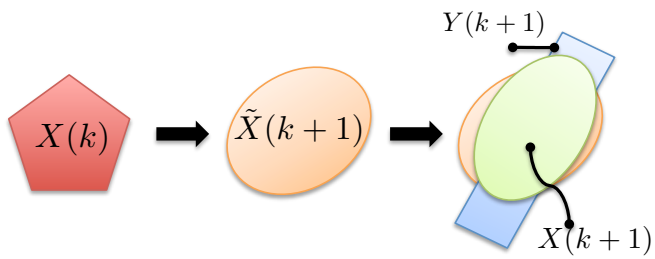


Fig. 7. Original set and ellipsoidal overbound with the set resulting from the intersection with the measurement set to form the new set-valued estimate.

- The matrix defining the polytope generally belongs to $\mathbb{R}^{\ell \times n_x}$, where $\ell \gg n_x$ and represents the number of restrictions associated with the edges of the polytope whereas the proposed overbound matrix belongs to $\mathbb{R}^{n_x \times n_x}$;
- Running the SVO computations only at a few time instants allows to use the idle moments to pre-compute the necessary combinations of matrices products;
- In some special cases of interest, such as in distributed systems, it is possible to discard dynamics matrices based on the observation set and compute the worst-case estimate with minimal processing effort.

The first rows of (2) are equivalent to

$$M(k)(A_0 + A_{\Delta^\star})^{-1}\mathbf{x} \leq m(k) \tag{13}$$

where (13) defines the set of points created by propagating the previous set defined by $\text{Set}(M(k), m(k))$. If the sets are always defined so as to have $m(k) = 1_{n_x}$, the state can be bounded at time k using Theorem 1. After computing the ellipsoidal overbound, it is propagated using the dynamics of the system, considering each given instantiation of the uncertainties in $\Delta^\star$. Thus, $X(k+1) = \{q : q = (A_0 + A_{\Delta^\star})^{-1}x \wedge x \in X(k)\}$.

From the previous discussion, the set $\tilde{X}(k+1)$ can be described as

$$\tilde{X}(k+1) = \text{co}(\bigcup_{\Delta \in \mathcal{H}} \text{Set}(M(k)(A_0 + A_\Delta(k))^{-1}, 1)$$

If the original set defined by matrix $M(k)$ is overbounded using the procedure found in Section III-B, the resulting overbound set $\tilde{X}(k+1)$ corresponds to the convex hull of the union of ellipsoids, where their axis correspond to the singular vectors basis (see the illustration in Fig. 7).

Due to sensor noise and/or the inability to measure the full state of the system, the observations can be defined as a set $Y(k)$ which is a polytope imposing constraints on the current state. The singular values and the associated singular vectors of the matrix defining such a polytope indicate the directions where the uncertainty is greater. Therefore, to test whether to execute the operation of computing the actual set, one can resort to intersecting the observation set $Y(k+1)$ with the ellipsoids and evaluate if the norm of the state increases over the current iteration to prevent it from becoming arbitrarily large. In other words, this allows us to derive conditions to decide whether to use the SVO procedure described in Section II or just update the overbound.

An easy-to-compute new estimate for the norm of the state at time $k+1$ can be obtained by solving

$$\begin{aligned} \text{maximize} \quad & ||p|| \\ \text{subject to} \quad & p^\intercal M(k+1)^\intercal M(k+1)p \leq 1 \\ & p \in Y(k+1). \end{aligned}$$

The previous problem translates into finding an intersection of ellipsoids and then computing the point in that set with the greatest norm. Matlab's Ellipsoidal Toolbox [39] provides computationally efficient methods to tackle this problem. The complexity associated with computing the intersection of two sets is constant in terms of required iterations and each is cubic in the dimension of the state, as it amounts to solve a Second-Order Cone Programming (SOCP).

A. Event-Triggered Set-Valued Observers

The description of how an Event-Triggered SVO works is similar to how an event-triggered system performs the sensor updates, as seen in Section IV-A. An event condition is based on requiring that the approximation ellipsoid is contained in the current maximum norm ball of radius $\mu(k)$. The value $\mu(k)$ is the minimum between a performance bound specified by the user and the last approximation set maximum norm, so as to guarantee convergence of estimates.

The Event-Triggered SVO computes an ellipsoid overbound that approximates the set $\tilde{X}(k)$ but which is *fast* to compute and fairly inexpensive when compared to some of the required computations of a "classical" SVO, such as the convex hull of (2) for each of the uncertainty instantiations. A *full* iteration of the SVO is going to be triggered at time $\tau^0(k)$ if the following does not hold

$$\mathcal{E}(\tau^0(k)) \subseteq \mathcal{B}(\mu(\tau^{-1}(k))) \tag{14}$$

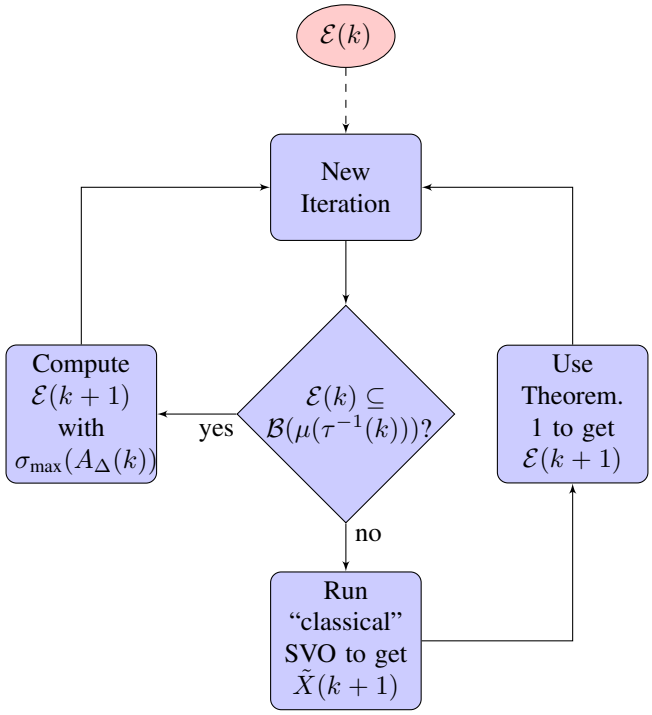


Fig. 8. Flowchart of the Self-Trigger SVO algorithm where $\mathcal{E}(k)$ and $\mathcal{B}_{\mu(k)}$ are the overbounding ellipsoid at time k and the ball of radius $\mu(k)$ centered at the origin, respectively.

where $\mathcal{E}(\tau^0(k))$ is the ellipoid approximation at the current time and $\mathcal{B}(\mu(\tau^{-1}(k)))$ is a ball centered at the origin of radius $\mu(\tau^{-1}(k))$. When triggered, the observer gets $\tilde{X}(\tau^0(k))$ using (2) and then using Theorem 1 obtain the new $\mathcal{E}(\tau^0(k))$. The new radius for the event condition is given by

$$\mu(\tau^0(k)) = \min(\mu_u, \frac{n_x}{\sigma^2_{\min}(M(\tau^{-1}(k)))}) \tag{15}$$

where μ_u is a user provided performance minimum. Notice that it is not possible to use $M(\tau^0(k))$ in (15) because if no measurement is available this would result in the mechanism triggering every instant. This algorithm is summarized in Fig. 8.

The event condition used for the SVOs is very similar to that of Section IV-A and enables the use of both strategies so as to avoid communications between the sensor and the observer. Additionally, the observer can return set-valued estimates without a heavy computational burden in between sensor updating times. In such a scenario, the SVO can output the sets $\mathcal{E}(\tau^{-1}(k)), \mathcal{E}(\tau^{-1}(k)+1), \cdots, \mathcal{E}(k)$, until $\mathcal{E}(k) \nsubseteq \mathcal{B}(\mu(\tau^{-1}(k)))$. The set $\mathcal{E}(k)$ increases in hyper-volume since there is no intersection with the measurement set. In summary, if both strategies were to be used together, the sensor would be testing whether its last observation is within the received set to trigger an event, whereas the observer is outputting the ellipsoidal sets for the worst-case until their hyper-volume is higher than the last trigger.

B. Self-Triggered Set-Valued Observers

Self-Triggered SVOs aim to have the important feature of allowing the process running the observer to be shut down during the time between each trigger. Nonetheless, the set-valued estimates for that period might be needed by some application, which is somehow contradictory. However, the observer can take advantage of the computations performed when it was determining the next trigger instant, where it computed the propagated set for all time instants from the current time until the next trigger. These sets can be used as the set-valued estimates in between updates.

The procedure is identical to the triggering mechanism detailed in Section IV-B, but using the update condition in (14). In essence, at each trigger time, the SVO will run the standard SVO iteration and obtain the polytope representing the set-valued estimate, and then compute the next trigger time. An inexpensive solution is to find the ellipsoidal approximation and propagate it in successive iterations until condition (14) is no longer satisfied. Doing so avoids computing the polytope sets for each of the time instants in between triggers.

The search for the next trigger time produces the set-valued estimates that are necessary for all the remaining future time instants in a lightweight fashion. As a consequence, there is no computation in between triggers. An interesting remark is that event- and self-triggered strategies can be combined at two different levels. As an example, the observer can be running a Self-Triggered SVO and, at each triggering time, outputting the estimates up to the next triggering time, while the sensors

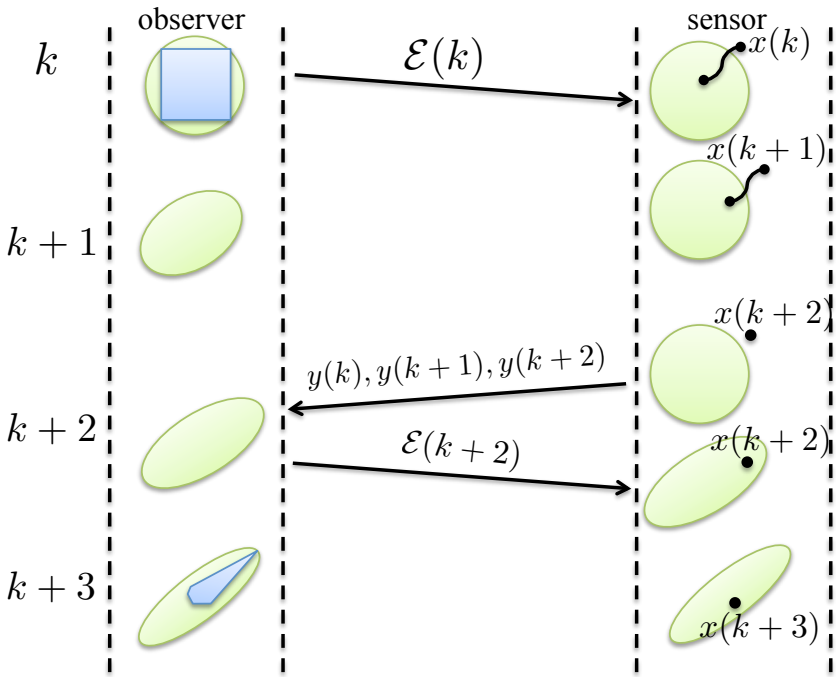


Fig. 9. Depiction of the observer and sensor sets for a combination of a Self-Triggered SVO used with an event-triggered NCS.

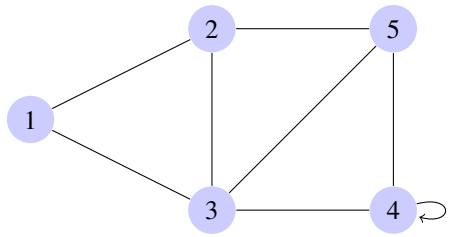


Fig. 10. Network example for a distributed system.

might be using that sequence of estimates to have an event-triggered sensor update. The processes requiring the estimates can intersect the event-based sensor updates with the self-triggered estimates.

Elaborating on the different levels where the techniques can be applied, the main difference between Self-Triggered SVOs and SVOs used for self-triggered systems relies on where the technique is applied. In the former, computational power is being saved at the observer level, by reducing the number of necessary state estimations using the standard SVO procedure. In the latter, the focus is on the network usage by the sensors in their updates.

The combination that is most advantageous is to have a Self-triggered SVO sending ellipsoidal approximations in between triggers to the sensor. Then, an Event-triggered strategy can be used at the sensor, testing whether the last received ellipsoid still contains the current measurement. In doing so, a communication only happens when the ellipsoid at the sensor does not include the current measurement. At this time instant, the sensor will send a batch of new measurements and expect to receive a new ellipsoid as state estimate. On the other hand, computational load is also reduced since a full SVO computation is triggered only when the current ellipsoid is not a suitable estimate. The full procedure will use the whole batch of measurements sent from the sensor since the last full computation produced all the ellipsoids that might be requested by the sensor in an event fashion (see Fig. 9 for a visual depiction of the interaction between sensor and observer).

C. Distributed Systems

The case of distributed systems is particularly relevant when considering fault detection applications. In particular, given the formulation in (1), it is possible to accommodate a distributed system by considering that each node is represented as a state and that the sequence of actions of each node defines the Δ parameters that selects a given overall system dynamics at any time instant. In Fig. 10, we depict an example of a network for a distributed system. Parameter Δ can represent, for instance, the communication between two nodes, i.e. the realization of the edges of the graph [38].

The definition also encompasses the case of distributed gossip algorithms where node selection and communication times are random processes. In this subsection, the systems to be considered satisfy the assumption that all dynamics matrices are

equal apart from a reordering of the rows and columns. The case of gossip algorithms fits this description in the sense that at each time instant a random pair of nodes performs some given operation using their states (see [32]). Such systems motivated the analysis of systems with the referred assumption which we formally introduce in the following definition

Definition 2 (reordering property): A dynamic system as in (1) has a *reordering property* in its nodes if the dynamics written as $A_0 + A_{\Delta_j}$, $\forall j : 1 < j \leq n_x$ satisfy

$$A_0 + A_{\Delta_j} = P(A_0 + A_{\Delta_1})P^\intercal, \forall j \tag{16}$$

where matrix P is a permutation matrix and j is any node different from 1.

Remark 2: Definition 2 can be found for example in the case of distributed gossip algorithms. However, more generally, one can have P being any orthogonal change of basis (i.e., $\forall j, A_0 + A_{\Delta_j} = P(A_0 + A_{\Delta_1})P^\intercal$ with $P \in \mathrm{O(n)}$).

Following Definition 2, let us introduce the following proposition which means that any ellipsoid set resulting from the propagation of the dynamics are the same up to a permutation change of basis.

Proposition 3: Take any ball $\mathcal{B}_b := \{q : \|q\| \leq b\}$ and matrices $A_0 + A_{\Delta_j}$, $1 \leq j \leq n_x$ as in Definition 2.

Then, all ellipsoids $\mathcal{E}_j := \{q : \frac{1}{b^2} x^\intercal (A_0 + A_{\Delta_j})^{-\intercal} (A_0 + A_{\Delta_j})^{-1} x \leq 1\}$ are equal to $\mathcal{E}_1$ up to a rotation.

Proof. From (16) and the fact that the permutation matrix is orthogonal (i.e., $P^\intercal P = I$), we get

$$\forall i : \sigma_i(A_0 + A_{\Delta_j}) = \sigma_i(A_0 + A_{\Delta_1}), \tag{17}$$

where $\sigma_i(A)$ is the ith singular value. We also have

$$\frac{1}{b^2} x^\intercal (A_0 + A_{\Delta_j})^{-\intercal} (A_0 + A_{\Delta_j})^{-1} x = \frac{1}{b^2} (P^\intercal x)^\intercal (A_0 + A_{\Delta_1})^{-\intercal} P^\intercal P (A_0 + A_{\Delta_1})^{-1} (P^\intercal x)$$

and due to (17)

$$\sigma(\frac{1}{b^2}(A_0 + A_{\Delta_1})^{-\intercal} P^\intercal P (A_0 + A_{\Delta_1})^{-1}) = \sigma(\frac{1}{b^2}(A_0 + A_{\Delta_1})^{-\intercal} (A_0 + A_{\Delta_1})^{-1})$$

with matrix P defining a rotation. Thus, the conclusion follows. ■

Proposition 3 allows the introduction of the following theorem that limits the amount of computations to verify if the norm exceeded the condition event, to that of testing if the singular vectors of each of the propagated ellipsoids aligns with the singular vectors of $Y(k)$.

Theorem 2: Consider a distributed algorithm with $\tilde{X}(0) = \{p : \|p\| \leq 1\}$, each dynamics matrix being written as $A_0 + A_{\Delta_j} = U_j S V_j^\intercal$, with v^j_{max} being the singular vector associated to the largest singular value and, conversely, an observation set $Y = \{U_y S_y V_y^\intercal p : \|p\|_\infty \leq 1\}$ with v^y_{max} corresponding to the largest singular value. Then, the worst-case set-valued state estimate overbound is given by the intersection of Y with the ellipsoid defined by Δ_j such that

$$\max_j (v^j_{\text{max}})^\intercal v^y_{\text{max}}$$

Proof. From Proposition 3 we get that all the ellipsoids computed using each of the parameters Δ_j are equal up to a rotation. Finding the largest intersection between any of these ellipsoids and Y can be found as the one for which the following optimization program has the highest solution:

$$\begin{aligned} \text{maximize} \quad & ||p|| \\ \text{subject to} \quad & p^\intercal M_j p \leq 1 \\ & p \in Y. \end{aligned}$$

where the $M_j = U_j S^{-2} V_j^\intercal$, following the conversion to ellipsoid representation. Since all matrices M_j are the same apart from a rotation, we are solving the equivalent problem

$$\begin{aligned} \text{maximize} \quad & ||p|| \\ \text{subject to} \quad & (R^\intercal p)^\intercal M_1 (R^\intercal p) \leq 1 \\ & p \in Y \\ & R \in \{R_1, \cdots, R_j\}. \end{aligned}$$

When R is unconstrained, has a closed-form solution given by $RV_y = V_1$ and the cost function evaluates to

$$\min(\sigma_{max}(S_y), \sigma_{max}(A_0 + A_{\Delta_1}))$$

i.e., the maximum singular vectors align. The optimization goal $\|p\|$ is monotonically increasing with the inner product of the maximum singular vectors of the ellipsoids and measurement set Y. Thus, the constrained version of the problem has a solution for the ellipsoid with the maximum inner product between its singular vectors and the singular vector of Y and the conclusion follows. ■

Theorem 2 establishes that the triggering condition presented in this paper is particularly effective in the context of distributed systems having the reordering property. In such systems, the worst case scenario can be found by checking the inner product

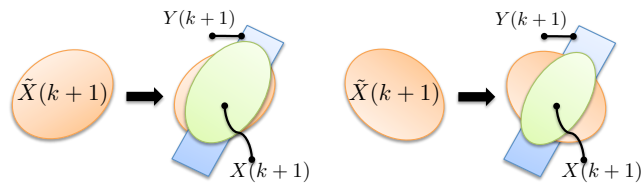


Fig. 11. Example demonstrating two ellipsoids and its corresponding intersection with the set of observations.

between the maximum singular vector of each possible dynamics matrix and the singular vectors of the observation set. This fact is illustrated in Fig. 11 where two ellipsoidal overapproximation sets are shown with the corresponding intersection with the set Y. As a consequence, at each time instant, the mechanism selects the overbounding ellipsoid producing the largest intersection. By doing so, the computational cost associated with the combinatorial behavior of the SVOs becomes constant given that only one ellipsoid needs to be computed along with one intersection and no unions are needed. This is irrespective of the number of dynamics matrices (i.e., the number of agents and states in the network).

VI. Triggering Frequency and Convergence

Self-triggered systems have the advantage of reducing the communication associated with the process at a frequency that depends on the characteristics of the system and its sensors. In this section, we analyze the triggering frequency by showing how the time until the next update can be inferred from the singular values of the system dynamics. We also demonstrate that the triggering techniques do not prevent estimate convergence of the standard SVO. For Event-triggered systems, such analysis cannot be performed as the trigger depends on the actual measurement value, but the results for Self-triggered can be viewed as a worst-case for the Event-triggered, in the sense that the condition for triggering corresponds to obtaining a measurement that is the *worst* possible, from the point of view of the stability of the set-valued estimates.

A. Worst-case Scenario

In this subsection, results regarding Problem 1 are presented. The next theorem gives an "easy-to-compute" alternative to the iterative testing of different triggering times introduced in this paper. Intuitively, the result shows how the size of the set-valued estimates relates to the system dynamics and allows to use an (off-line) pre-computed sub-optimal value for the triggering time.

Theorem 3: Consider a Self-Triggered SVO, as in Section V-B, with maximum state norm at the trigger time $\tau^{-2}(k)$ given by $||x(\tau^{-2}(k))|| \leq \mu(\tau^{-2}(k))$. The next trigger $\tau^1(k)$ occurs after $\mathcal{T}_k := \tau^1(k) - k$ time instants, where

$$\mathcal{T}_k = \left\lceil \log_\gamma \sigma_{\min}(M(k)) \frac{\mu(\tau^{-2}(k))(1-\gamma) - \sqrt{n}_d}{\sqrt{n}_x(1-\gamma) - \sqrt{n}_d \sigma_{\min}(M(k))} \right\rceil$$

with

$$\gamma = \max_{i \in \{1,\cdots,n_\Delta\}} \sigma_{\max}(A_0 + A_{\Delta_i})$$

Proof. Any point $x_1 \in X(k+1)$ satisfies $x_1 = (A_0 + A_\Delta)x_0$ for some $x_0 \in X(k)$ and some realization of the uncertainties Δ. Since it is assumed that the self-triggering technique translates, at each time instant, the set to incorporate the control law $B(k)u(k)$, then,

$$\begin{aligned}
||x(k+\mathcal{T}_k)|| =& ||(A_0 + A_{\Delta_{k+\mathcal{T}_k}})(A_0 + A_{\Delta_{k+\mathcal{T}_k-1}}) \cdots (A_0 + A_{\Delta_k})x(k) \\
&+ (A_0 + A_{\Delta_{k+\mathcal{T}_k-1}}) \cdots (A_0 + A_{\Delta_k})L(k)d(k) + \cdots + L(k+\mathcal{T}_k)d(k+\mathcal{T}_k)|| \\
\leq& \gamma^{\mathcal{T}_k}||x(k)|| + \sum_{j=0}^{\mathcal{T}_k-1} \gamma^j \sqrt{n_d} \\
\leq& \gamma^{\mathcal{T}_k} \frac{\sqrt{n}}{\sigma_{\min}(M(k))} + \sqrt{n_d}\frac{1-\gamma^{\mathcal{T}_k}}{1-\gamma}
\end{aligned}$$

which holds for the non-trivial case of $\gamma \neq 1$. To maintain the norm bounded $||x(k+\mathcal{T}_k)|| \leq \mu(\tau^{-2}(k))$ it thus required that

$$\begin{aligned}
&\gamma^{\mathcal{T}_k}\left(\frac{\sqrt{n}_x}{\sigma_{\min}(M(k))} - \frac{\sqrt{n}_d}{1-\gamma}\right) \leq \mu(\tau^{-2}(k)) - \frac{\sqrt{n}_d}{1-\gamma} \\
&\quad \Leftrightarrow \\
&\gamma^{\mathcal{T}_k} \leq \frac{\mu(\tau^{-2}(k)) - \frac{\sqrt{n}_d}{1-\gamma}}{\frac{\sqrt{n}_x}{\sigma_{\min}(M(k))} - \frac{\sqrt{n}_d}{1-\gamma}} \\
&\quad \Leftrightarrow \\
&\mathcal{T}_k \leq \log_\gamma \sigma_{\min}(M(k)) \frac{\mu(\tau^{-2}(k))(1-\gamma) - \sqrt{n}_d}{\sqrt{n}_x(1-\gamma) - \sqrt{n}_d \sigma_{\min}(M(k))}
\end{aligned}$$

thus, leading to the conclusion. ■

Theorem 3 presented a relationship between the system dynamics and the triggering frequency. An analogous result can be derived for the case of a system where the dynamics are selected from a set of possible matrices according to a stochastic variable. In such a setup, the probability distribution for the parameter Δ is known. Practical examples of this model range over distributed stochastic systems where some decision is random or the nodes acting at a given time instant are stochastically chosen.

An important issues when introducing such a technique is its impact in the convergence of the estimates. We recall for completeness the result proved in [18] regarding the boundedness of the produced sets in terms of hypervolume.

Proposition 4: Suppose that a system described by (1) with $x(0) \in X(0)$ and $u(k) = 0, \forall k$, satisfies, for sufficiently large $N^\star$,

$$\gamma_N := \max_{\Delta(k),\cdots,\Delta(k+N)} \left\| \prod_{j=k}^{k+N} \left[A_0 + \sum_{i=1}^{n_\Delta} \Delta_i(j) A_i \right] \right\| < 1,$$

for all $N \geq N^\star$. Then, $\tilde{X}(k)$ cannot grow without bound.

Based on the condition in Proposition 4, we can introduce the counterpart for the convergence with the triggering schemes.

Proposition 5: Suppose that a system described by (1) satisfies Proposition 4 for a given $N^\star$. Then, the following conditions are satisfied:

i) A Self-Triggered SVO cannot grow without bound by considering $N \geq N^\star$;
ii) A Self-Triggering System using an SVO has estimates that cannot grow without bound by considering $N \geq N^\star$.

Proposition 5 comes directly from the fact that no assumptions are required in Proposition 4 regarding the measurements. In both cases there is, at some point in time, a computation of the standard set-valued estimates using SVOs. Proposition 4 takes into consideration only the dynamics of the system. The next theorem presents a similar result incorporating the effect of the intersection with the measurement set. The intuition behind the result is that in the directions that the system is measured, the requirement for stability can be dropped, as the intersection will decrease the uncertainty in those directions.

Theorem 4 (SVO convergence): Suppose that a system described by (1) with $x(0) \in X(0)$ and $u(k) = 0, \forall k$, verifies, for sufficiently large $N^\star$,

$$\gamma_N := \max_{\Delta(k),\cdots,\Delta(k+N)} \left\| \text{null}(C(k+N)) \prod_{j=k}^{k+N} \left[A_0 + \sum_{i=1}^{n_\Delta} \Delta_i(j) A_i \right] \right\| < 1 - \delta_N,$$

for all $N \geq N^\star$, where null$(C(k+N))$ is the matrix defining a null space orthonormal basis of $C(k+N)$ and

$$\delta_N := \max_{d(k),\cdots,d(k+N-1)} \|A_k^{N-1} L(k) d(k) + \cdots + L(k+N-1) d(k+N-1)\|.$$

Then, the hypervolume of $\tilde{X}(k)$ is bounded.

Proof. Consider the ellispoidal overbound given by Theorem 1 for the state at time k, denoted by $\mathcal{E}(k)$, where without loss of generality $\mathcal{E}(k) = \{x : \|x\| \leq 1\}$. The maximum norm of any point belonging to any $\mathcal{E}(k+N)$ satisfies

$$\begin{aligned}
\|x(k+N)\| &\leq \gamma_N \|x(k)\| + \delta_N \\
&\leq \gamma_N + \delta_N
\end{aligned}$$

since the intersection along each of the directions in $C(k+N)$ is at most $2\nu^\star$ as it is the size of $Y(k+N)$.

If $N \geq N^\star$, $\|x(k+N)\| \leq 1$ implying $\mathcal{E}(k+N) \in \mathcal{E}(k)$ and additionally $\tilde{X}(k+N) \in \mathcal{E}(k)$ and the conclusion follows. ■

Theorem 4 refines the result in Proposition 4 by noticing two facts: we can discard the directions associated with the measurement matrix $C(\cdot)$ since the maximum size of the intersection with $Y(k)$ is going to be $2\nu^\star$; and, in the worst-case, the decrease in norm associated with the dynamics compensates the increase associated with the disturbance signal.

B. Stochastic case

Another case of interest is to analyze the triggering frequency when the probability distribution for the uncertain parameter Δ of matrix $A(\Delta(k))$ is known (i.e., in the context of Problem 2). Before stating the result, the following definitions are required, where $\inf$, $\sup$ and $\varnothing$ denote respectively the infimum, supremum and empty set.

Definition 3 (volume expansion stochastic variable): For a distributed system where the dynamics $A(\Delta(k))$ is selected from a set $\{A_0 + A_{\Delta_i} : 1 \leq i \leq n_\Delta\}$ following a probability distribution where A_{Δ_i} is picked with probability p_i, define the sequence of *volume expansion stochastic variables* as $\theta(k) = \sigma_{\max}(A(\Delta(k)))$, with probability p_i.

Notice that the stochastic variable for the volume expansion is the stochastic equivalent of the quantity γ in Theorem 3. The results in this section only assume the knowledge of the expected value of the distribution and not on the distribution itself since we are focusing on the expected value for the triggering frequency. For an example assuming a particular probability distribution see [40].

Definition 4 (upcrossing): For a sequence of stochastic variables $Z_1, \cdots, Z_n$ and two real numbers a and b, define

$$S_{k+1}(Z) = \inf\{n \geq T_k(Z) : Z_n \leq a\} \text{ and } T_{k+1}(Z) = \inf\{n \geq S_{k+1}(Z) : Z_n \geq b\}$$

with the usual convention that $\inf \varnothing = \infty$. The number of upcrossings of the sequence Z of the interval $[a, b]$ in n time instants is defined as

$$U_n([a, b], Z) = \sup\{k \geq 0 : T_k(Z) \leq n\}.$$

Notice that the definition of upcrossing in Definition 4 of random variables is going to be equivalent to a trigger in our application. The volume exceeding the triggering condition corresponds to the random variable corresponding to that volume making an upcrossing of the interval. We now introduce the theorem stating the results for the triggering frequency in randomized algorithms, where $\mathbb{E}$ denotes the expected value operator and $\mathbb{P}$ the probability function.

Theorem 5: Consider a distributed system and a stochastic variable as in Definition 3. Then,

i) if $\mathbb{E}[\theta(n)] < 1$, the volume of the set-valued estimates converges *almost surely* to a nonnegative integrable limit and $\mathbb{P}[\text{“having a trigger”}] \leq \frac{\mu(k)}{\mu(\tau^{-2}(k))}$;

ii) if $\mathbb{E}[\theta(n)] = 1$, the expected triggering time is given by $\mathbb{E}[\mathcal{T}_k|k] = \log \mu(k) \log \frac{\mu(\tau^{-2}(k))}{\mu(k)}$, where $\mathcal{T}_k := \tau^1(k) - k$;

iii) if $\mathbb{E}[\theta(n)] > 1$, the time before the expected value of the number of triggers is greater than or equal to 1 is given by the M that satisfies $\mathbb{E}[|Z_M - Z_0|] < \mu(\tau^{-2}(k)) - \mu(\tau^{-1}(k))$.

Proof. i) Consider the stochastic process $Z_{n+1} = Z_n\theta(n)$, $Z_0 = \mu(k)$ describing the behavior of the size of the set-valued estimates for the distributed system. Also consider the correspondent filtration $\mathcal{F}_n = \{Z_0, Z_1, \cdots, Z_n\}$ (for additional information on martingale theory, see [41]).

Computing the conditional expectation, we get

$$\begin{aligned}
\mathbb{E}[Z_{n+1}|\mathcal{F}_n] &= \mathbb{E}[\theta(n)Z_n|\mathcal{F}_n] \\
&= Z_n\mathbb{E}[\theta(n)|\mathcal{F}_n] \\
&= Z_n\mathbb{E}[\theta(n)] \\
&< Z_n
\end{aligned}$$

which implies Z_n is a nonnegative supermartingale and $\lim_{n\to\infty} Z_n = Z_\infty$ with Z_∞ being a a nonnegative integrable variable as stated in page 148 of [42]. The final conclusion is a direct application of the Dubin's Inequality that states

$$\mathbb{P}[\text{“at least } \zeta \text{ upcrossings”}] \leq \left(\frac{a}{b}\right)^\zeta$$

where a and b define upcrossings as in Definition 4. In our context, a is the initial value and b represents the maximum volume before a trigger. Thus, the number of upcrossings is the number of triggers. As a consequence, if Z_n reaches 0 it must stay there forever.

ii) Consider the above martingale $Z_{n+1} = Z_n\theta(n)$, $Z_0 = \mu(k)$ and define the new martingale $V_n = \log Z_n$. Therefore, we have that $V_{n+1} = V_n + \xi(n)$, where $\xi(n)$ is $\log\theta(n)$, along with the correspondent filtration $\mathcal{F}_n = \{V_0, V_1, \cdots, V_n\}$.

Computing the conditional expectation, we get

$$\begin{aligned}
\mathbb{E}[V_{n+1}|\mathcal{F}_n] &= \mathbb{E}[V_n + \xi|\mathcal{F}_n] \\
&= \mathbb{E}[V_n|\mathcal{F}_n] + \mathbb{E}[\xi|\mathcal{F}_n] \\
&= V_n + \mathbb{E}[\xi].
\end{aligned}$$

By definition, $\mathbb{E}[\xi(n)] = \log\mathbb{E}[\theta(n)]$ which implies that $\mathbb{E}[V_{n+1}|\mathcal{F}_n] = V_n$ and indeed V_n is a martingale. We progress by writing the stochastic variable $W_n = V_n^2 - n$ and showing that it can indeed be made a martingale. Take the correspondent

filtration $\mathcal{F}_n = \{V_0, V_1, \cdots, V_n\}$ and let us compute

$$\begin{aligned}\mathbb{E}[W_{n+1}|\mathcal{F}_n] &= \mathbb{E}[V_{n+1}^2 - (n+1)|\mathcal{F}_n] \\ &= \mathbb{E}[V_n^2 + 2V_n\xi + \xi^2 - (n+1)|\mathcal{F}_n] \\ &= V_n^2 + 0 + \mathbb{E}[\xi^2|\mathcal{F}_n] - (n+1).\end{aligned}$$

Without loss of generality, we assume variable ξ to have the expected value of its square equal to 1. This can be achieved by scaling the state of the system. Thus, simplifying to

$$\begin{aligned}\mathbb{E}[W_{n+1}|\mathcal{F}_n] &= V_n^2 - n \\ &= W_n\end{aligned}$$

Let us consider the stopping time corresponding to our self-triggering techinque

$$\mathcal{T}_k = \inf\left\{n \geq 0 : V_n = \log\mu(\tau^{-2}(k))\right\}$$

Due to the martingale properties, $V_{\mathcal{T}_k \wedge n}$ is a martingale which implies that

$$\mathbb{E}[V_{\mathcal{T}_k \wedge n}|k] = \mathbb{E}[V_{\mathcal{T}_k \wedge 0}|k] = \mathbb{E}[V_0|k] = \log\mu(k).$$

We can also compute the probability of hitting the maximum volume $\log\mu(\tau^{-2}(k))$ by computing

$$\begin{aligned}&\mathbb{E}[V_{\mathcal{T}_k}|k] = \log\mu(\tau^{-2}(k))\mathbb{P}[V_{\mathcal{T}_k} = \log\mu(\tau^{-2}(k))|k] \\ &\Leftrightarrow \\ &\mathbb{E}[V_0|k] = \log\mu(\tau^{-2}(k))\mathbb{P}[V_{\mathcal{T}_k} = \log\mu(\tau^{-2}(k))|k] \\ &\Leftrightarrow \\ &\mathbb{P}[V_{\mathcal{T}_k} = \log\mu(\tau^{-2}(k))|k] = \frac{\log\mu(k)}{\log\mu(\tau^{-2}(k))}.\end{aligned}$$

Using the new martingale

$$\mathbb{E}[W_{\mathcal{T}_k \wedge n}|k] = \mathbb{E}[W_{\mathcal{T}_k \wedge 0}|k] = \mathbb{E}[W_0|k] = (\log\mu(k))^2, \tag{18}$$

and also

$$\mathbb{E}[W_{\mathcal{T}_k \wedge n}|k] = \mathbb{E}[V_{\mathcal{T}_k \wedge n}^2 - \mathcal{T}_k \wedge n|k] \tag{19}$$

Using both (18) and (19), we get

$$\mathbb{E}[V_{\mathcal{T}_k \wedge n}^2|k] = (\log\mu(k))^2 + \mathbb{E}[\mathcal{T}_k \wedge n|k]. \tag{20}$$

Due to the Monotone Convergence theorem, as $\mathcal{T}_k \wedge n \to \mathcal{T}_k$, $\mathbb{E}[\mathcal{T}_k \wedge n|k] \to \mathbb{E}[\mathcal{T}_k|k]$, which combined with (20) leads to

$$\mathbb{E}[V_{\mathcal{T}_k}^2|k] = (\log\mu(k))^2 + \mathbb{E}[\mathcal{T}_k|k] \tag{21}$$

but by definition

$$\mathbb{E}[V_{\mathcal{T}_k}^2|k] = (\log\mu(\tau^{-2}(k)))^2\mathbb{P}[V_{\mathcal{T}_k} = \log\mu(\tau^{-2}(k))|k] = (\log\mu(\tau^{-2}(k)))^2\frac{\log\mu(k)}{\log\mu(\tau^{-2}(k))} \tag{22}$$

Using (21) and (22), we get that $\mathbb{E}[\mathcal{T}_k|k] = \log\mu(k)\log\frac{\mu(\tau^{-2}(k))}{\mu(k)}$, thus reaching the conclusion.

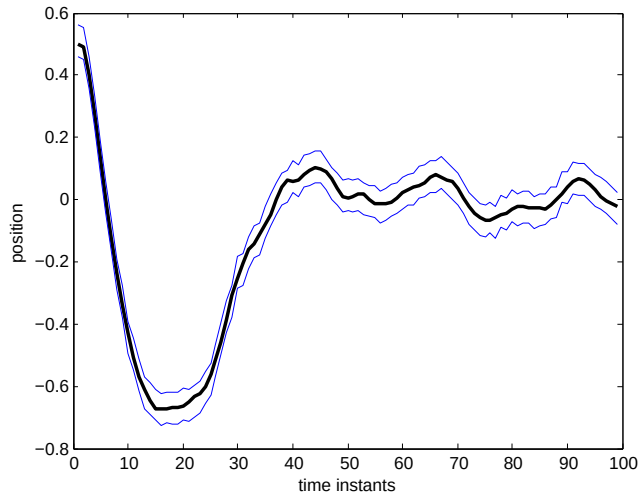
iii) We consider the submartingale Z_n and recall the Upcrossing Lemma that states

$$\mathbb{E}[U_M^{\alpha,\beta}] \leq \frac{\mathbb{E}[|Z_M - Z_0|]}{\beta - \alpha}$$

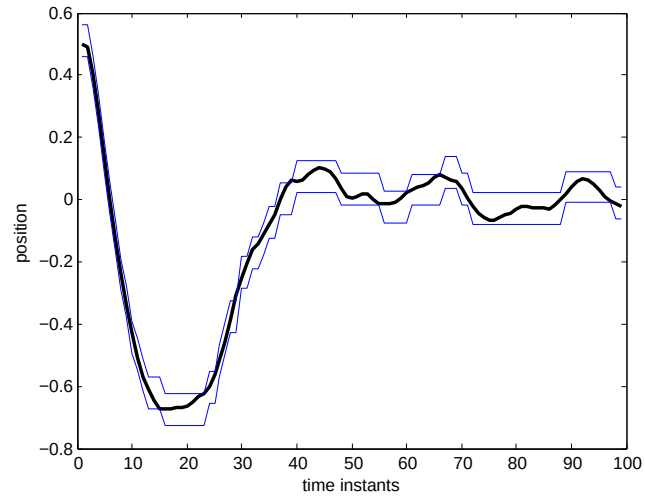
for a submartingale Z_n. To get $\mathbb{E}[U_M^{\alpha,\beta}] < 1$, we must get

$$\frac{\mathbb{E}[|Z_M - Z_0|]}{\beta - \alpha} < 1$$

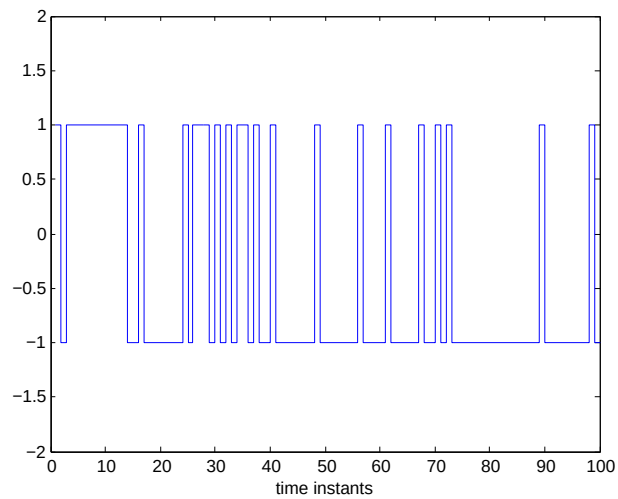
which is satisfied by selecting M as in the statement of the theorem where α is the current hypervolume of the set-valued estimates and β is the maximum allowed hypervolume and the conclusion follows. ■



(a) Evolution of the position of the cart (black) and the estimation (blue) of the standard SVO.



(b) Estimation (blue) given by the standard SVO in a event-triggered sensor update scheme.



(c) Triggering occurrences at around a third of the time instants.

Fig. 12. Estimation conservatism and triggering frequency of the event-triggering strategy for NCS using the standard SVOs.

VII. Simulation Results

In this section, we start by illustrating the advantages of the proposed event- and self-triggering techniques in order to reduce sensor updates. We consider a linearized model of the inverted pendulum mounted on a cart, which relates directly to the real-world example of an attitude control of a booster rocket at takeoff. In continuous time, the state dynamics are given by

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\\ \dot{\theta}\\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & \frac{-(I+m\ell^2)b}{I(M+m)+Mm\ell^2} & \frac{m^2g\ell^2}{I(M+m)+Mm\ell^2} & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{-m\ell b}{I(M+m)+Mm\ell^2} & \frac{mg\ell(M+m)}{I(M+m)+Mm\ell^2} & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \theta\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ \frac{I+m\ell^2}{I(M+m)+Mm\ell^2}\\ 0\\ \frac{m\ell}{I(M+m)+Mm\ell^2}\end{bmatrix} u + Lw$$

$$y = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \theta\\ \dot{\theta}\end{bmatrix} + N\nu$$

where x is the cart position coordinate and θ is the pendulum angle from vertical. The constants appearing in the model are the moment of inertia of the pendulum ($I = 0.006$ kg.m^2), length to pendulum center of mass ($\ell = 0.3$m), coefficient of friction for cart ($b = 0.1$ N/m/sec), mass of the pendulum ($m = 0.2$kg), and mass of the cart ($M = 0.5$kg). The system is discretized using a sampling period of 0.1s.

We assume a matrix L for the disturbances equal to two and a half times the input matrix as to make the problem harder by having a large disturbance signal, and the noise injection matrix N is $\begin{bmatrix}0.5 & 0.5\end{bmatrix}^\intercal$. The random signal w is taken from a normal distribution with variance equal to 1 and mean 0 with a maximum imposed of at most 5, which represents a large disturbance signal when compared to the control input. The control law u is assumed to be given by a state-feedback controller, independent from the SVO, that returns a signal which stabilizes the unperturbed system. The objective is for the SVO in a NCS to provide estimates for the state of the remote system comprised of both the controller and the plant.

The first simulations focus on showing the properties of using SVOs to produce event conditions for the sensor to determine when to perform an update and send information through the network to the observer, i.e, SVOs for Event-triggered NCS as in Section IV-A. Figure 12 presents the main results of using an event condition based on the produced set-valued estimates of the state. In Fig. 12(b), it is depicted the interval for the values of the state that the observer outputs with the sensor updating according to the signal in Fig. 12(c) as opposed to having sensors updates at every time instant, which would result in the estimates given in Fig. 12(a). The main observation is that the technique does not introduce conservatism in the estimates as the observer makes them constant within triggers as the current observations are *validated* by the sensors as otherwise, an update would take place.

The event-triggered for NCS strategy simulated in Fig. 12 showed that for the considered system, the triggering frequency is close to one third of the time instances. Such a reduction motivates our contribution of using SVOs to determine triggering strategies as considerable load on the networked would be avoided in comparison with the standard approach of receiving measurements in all time instants.

The event-triggering strategy required the sensor unit to test whether the measurements are still inside the provided event condition set. We simulated the self-triggered version as to compare the results, i.e., SVOs for Self-triggered NCS as in Section IV-B. In Fig. 13, it is depicted the same results for a different run of the algorithm but still allowing to point out the trade-off between both strategies. Contrarily to the event-triggered condition, it is observed in Fig. 13(b) that the size of the estimation set varies due to some conservativeness being introduced by not having access to the sensor update. However, the convergence properties of the SVOs are maintained, since upon a trigger, the standard procedure is executed.

Figure 13(c) shows the frequency of the triggers with approximately 60% of the time instants having a trigger for the standard procedure. The main reason is the large disturbance and noise signals that make the produced sets grow in hyper-volume when no measurement is available. In essence, to have the possibility to switch off sensors in a self-triggered strategy, for this scenario, there is a two-fold increase in the number of triggers and a poorer estimate quality. Nevertheless, the contribution of using SVOs to self-trigger NCSs should be seen for a different use when the sensors cannot be equipped with any type of computational capabilities and all operations must be performed at the observer side. A saving of roughly $\frac{1}{3}$ of the network resources associated with communication is still encouraging.

A third simulation is performed resorting to the same example but for the Self-triggered SVOs as in Section V-B. In the previous cases, all computations were performed resorting to the traditional SVOs whenever sensors updates were available. In this simulation, sensor updates are available at every time instant and triggers mean that the standard SVOs were computed.

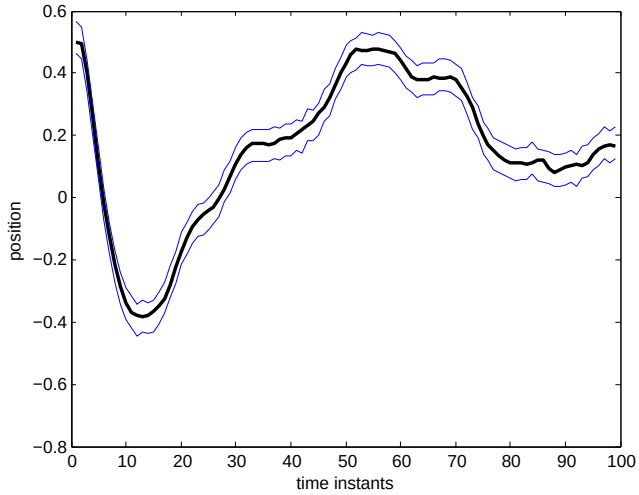
The results are shown in Fig. 14. For the run depicted in Fig. 14(a), we have the computed set-valued estimates in Fig. 14(b) using the overbounding methods as aforementioned. A main difference is the introduced conservatism due to the ellipsoidal overbounding method, which is worsen by propagating for all possible values of the disturbance and noise signal. The triggering frequency was around 50% in this run as shown in Fig. 14(c). The main conclusion from this simulation is that the self-trigger SVO can be an alternative to the traditional especially in cases where the disturbance and noise signals have a small magnitude. When that is not the case, this example gives evidence that the triggering frequency is high given that the set-valued estimates after a trigger are never of such a small volume to avoid a considerable increase in overhead.

Nevertheless, a Self-Trigger SVO can expand the class of systems to which the SVOs can be applied. A special consideration are always systems that need to be discretized with small sampling periods or real-time plants. In those cases, time constraints are of utmost important and place strict performance lower bounds on any potential technique. In Fig. 15, it is depicted the computing time for the traditional and Self-trigger SVOs. The minimum computational time for the traditional SVO was 3.6×10^{-2}s in contrast with 1.6×10^{-4}s for the proposed upperbounding method, representing a decrease of two orders of magnitude. Thus, Self-triggered SVOs can be employed for system with stricter time constraints as long as the full computations at triggering times can be performed in between triggers, suggesting some future work in this area.

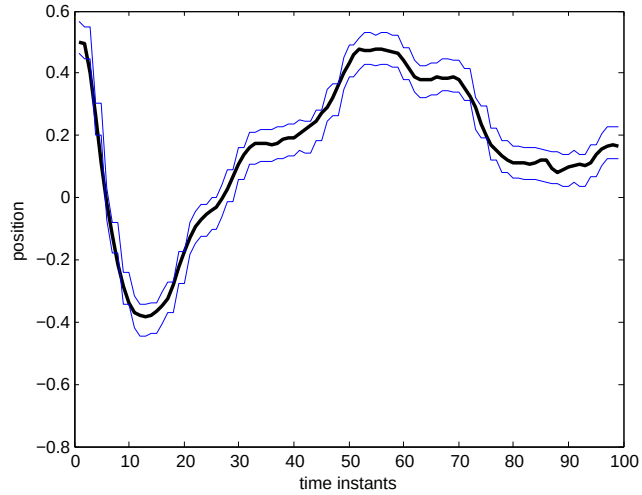
VIII. Conclusions

In this paper, the problem of reducing the network load in a Networked Control System (NCS) was addressed resorting to event- and self-triggered strategies. For this purpose, the concept of SVOs was used to provide polytopes where the state is known to belong and the triggering condition is selected such that the hyper-volume of the set-valued estimates does not grow. The algorithm does not impact on the convergence of the estimates since when the estimates increase in size, a sensor update is required to reduce the uncertainty in the estimates.

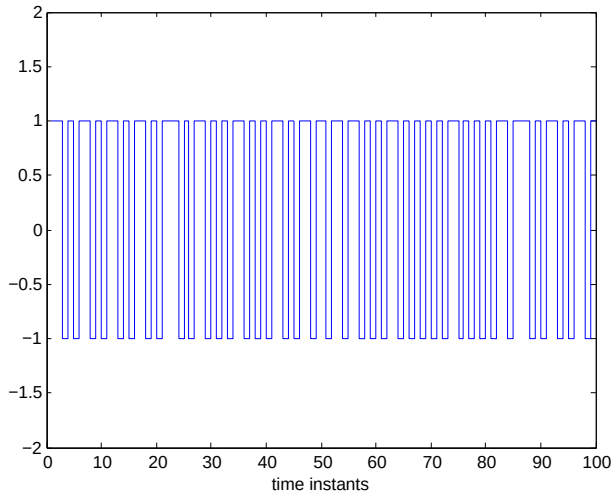
Following the study of triggering techniques, we provided similar event conditions to determine when to run a full computation of the SVOs to find the polytope for the estimates or compromise accuracy to gain in performance by giving low-complexity



(a) Evolution of the position of the cart (black) and the estimation (blue) of the standard SVO.



(b) Estimation (blue) given by the standard SVO in a self-triggered sensor update scheme.



(c) Triggering occurrences at around 60% of the time instants.

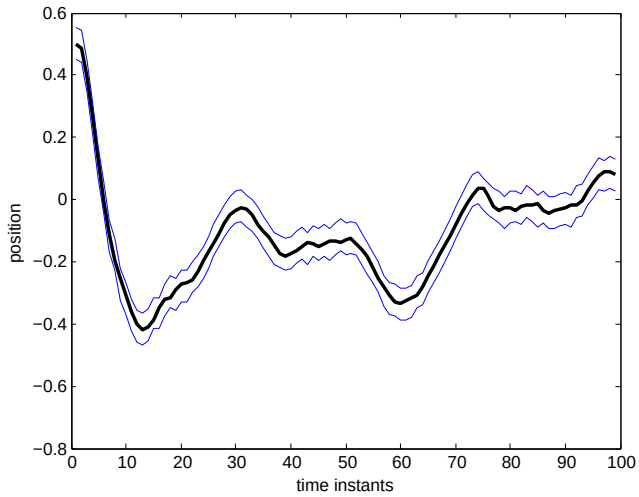
Fig. 13. Estimation conservatism and triggering frequency of the self-triggering strategy for NCS using the standard SVOs.

hyper-parallelepiped and ellipsoidal overapproximations. These were obtained by introducing an algorithm to find a centrally symmetric polytope for the estimates. This was guaranteed to not deteriorate the approximations and by performing a rotation, the case of polytopes defined with ill-conditioned matrices, which can have arbitrarily bad approximations, was shown to introduce an error factor that grows with the factorial of the state space.

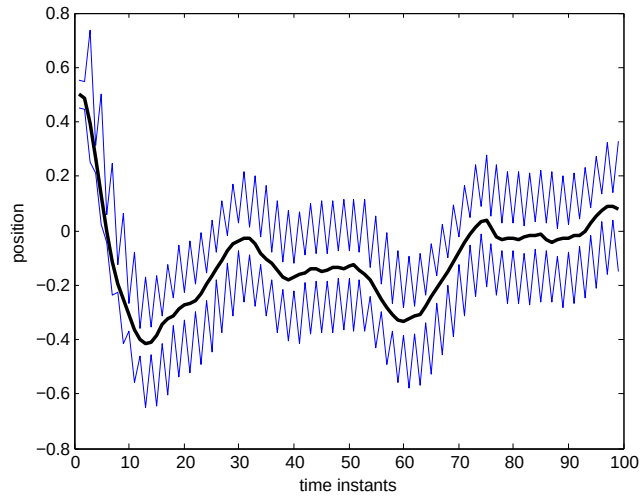
The convergence of the proposed strategies was shown as long as the conditions for the convergence of SVOs are satisfied by the system. In this paper, it was presented a novel result for convergence that generalizes an existing result in the literature by noticing that along the directions associated with the rows of matrix $C(\cdot)$, the maximum size of the polytope is always at most $2\nu^\star$ and therefore is independent of the dynamics of the system.

The triggering frequency was studied and shown to depend on the maximum singular value of the possible dynamics matrix and the minimum singular value of the matrix defining the polytope, which *measures* the maximum norm of the previous estimation set. The case of distributed systems was also addressed when the probability distribution for the dynamics matrix is known for the cases where the expected value for the maximum singular value is smaller, equal to or greater than one.

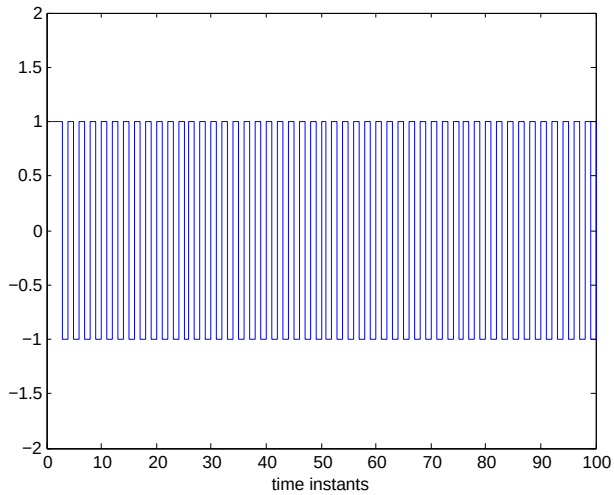
The work presented in this document suggests a natural course for future developments. Two main avenues of research will be pursued: how to extend the event- and self-triggering techniques here described to other set-based methods and what additional results can be provided for different set descriptions; and, whether alternative optimization techniques exist that can be employed to determine the next self-triggered time instant apart from generating all the ellipsoids and checking whether they satisfy the triggering criteria.



(a) Evolution of the position of the cart (black) and the estimation (blue) of the standard SVO.



(b) Estimation (blue) given by the Self-triggered SVO.



(c) Triggering occurrences at half of the time instants.

Fig. 14. Estimation conservatism and triggering frequency of the Self-triggered SVOs in comparison with the standard SVOs.

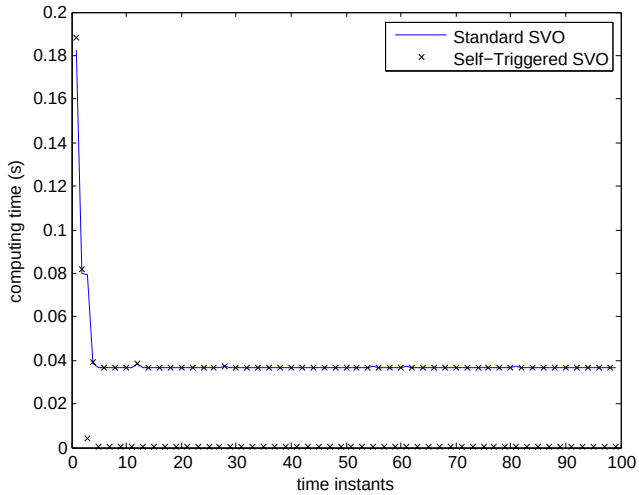


Fig. 15. Elapsed time in seconds of the computation of the estimates using the standard and Self-triggered SVOs.

REFERENCES

[1] G. C. Walsh, H. Ye, and L. G. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, May 2002.

[2] X. Zhao, C. Ma, X. Xing, and X. Zheng, "A stochastic sampling consensus protocol of networked euler-lagrange systems with application to two-link manipulator," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 907–914, Aug 2015.

[3] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, vol. 11, no. 10, pp. 1099 – 1111, 2003, special Section on Control Methods for Telecommunication. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066103000364

[4] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1109/jproc.2006.887288

[5] X. M. Zhang, Q. L. Han, and X. Yu, "Survey on recent advances in networked control systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1740–1752, Oct 2016.

[6] X. Ge, F. Yang, and Q.-L. Han, "Distributed networked control systems: A brief overview," *Information Sciences*, vol. 380, pp. 117 – 131, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025515005551

[7] A. Bemporad, M. Heemels, and M. Johansson, *Networked control systems*. Springer, 2010, vol. 406.

[8] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control." in *51st IEEE Conference on Decision and Control*, 2012, Maui, HI, USA,, pp. 3270–3285.

[9] H. Witsenhausen, "Sets of possible states of linear systems given perturbed observations," *IEEE Transactions on Automatic Control*, vol. 13, no. 5, pp. 556 – 558, oct 1968.

[10] F. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 22 – 28, feb 1968.

[11] F. Schweppe., *Uncertain Dynamic Systems*. Prentice-Hall, 1973.

[12] M. Milanese and A. Vicino, "Optimal estimation theory for dynamic systems with set membership uncertainty: An overview," *Automatica*, vol. 27, no. 6, pp. 997 – 1009, 1991.

[13] P. Casau, P. Rosa, and C. Silvestre, "FITBOX - a Fault Isolation Toolbox," *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 283 – 288, 2015, 9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS.

[14] D. Bertsekas and I. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *IEEE Transactions on Automatic Control*, vol. 16, no. 2, pp. 117 – 128, apr 1971.

[15] C. Combastel, "A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes," in *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05.*, dec. 2005, pp. 7228 – 7234.

[16] T. Alamo, J. Bravo, and E. Camacho, "Guaranteed state estimation by zonotopes," *Automatica*, vol. 41, no. 6, pp. 1035 – 1043, 2005.

[17] R. E. Moore, *Interval analysis*, ser. Prentice-Hall series in automatic computation. Englewood Cliffs, NJ: Prentice-Hall, 1966.

[18] P. Rosa and C. Silvestre, "Fault detection and isolation of LPV systems using set-valued observers: An application to a fixed-wing aircraft," *Control Engineering Practice*, vol. 21, no. 3, pp. 242 – 252, 2013.

[19] Y. Wei, J. Qiu, H. R. Karimi, and M. Wang, "Filtering design for two-dimensional markovian jump systems with state-delays and deficient mode information," *Information Sciences*, vol. 269, pp. 316 – 331, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025513008980

[20] D. Antunes and W. Heemels, "Rollout event-triggered control: Beyond periodic control performance," *Automatic Control, IEEE Transactions on*, vol. 59, no. 12, pp. 3296–3311, Dec 2014.

[21] J. Almeida, C. Silvestre, and A. Pascoal, "Self-triggered output feedback control of linear plants in the presence of unknown disturbances," *Automatic Control, IEEE Transactions on*, vol. 59, no. 11, pp. 3040–3045, Nov 2014.

[22] ——, "Self-triggered state-feedback control of linear plants under bounded disturbances," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 8, pp. 1230–1246, 2015. [Online]. Available: http://dx.doi.org/10.1002/rnc.3138

[23] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *Automatic Control, IEEE Transactions on*, vol. 56, no. 10, pp. 2456–2461, Oct 2011.

[24] ——, "On event-triggered and self-triggered control over sensor/actuator networks," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, Dec 2008, pp. 435–440.

[25] W. P. M. H. Heemels, J. H. Sandee, and P. P. J. V. D. Bosch, "Analysis of event-driven controllers for linear systems," *International Journal of Control*, vol. 81, no. 4, pp. 571–590, 2008. [Online]. Available: http://dx.doi.org/10.1080/00207170701506919

[26] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601 – 611, 1976.

[27] I. Hwang, S. Kim, Y. Kim, and C. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *Control Systems Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 636 –653, may 2010.

[28] J. Bokor and G. Balas, "Detection filter design for LPV systems – a geometric approach," *Automatica*, vol. 40, pp. 511–518, 2004.

[29] G. Ducard, *Fault-tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles*, ser. Advances in industrial control. Springer, 2009.

[30] S. Narasimhan, P. Vachhani, and R. Rengaswamy, "New nonlinear residual feedback observer for fault diagnosis in nonlinear systems," *Automatica*, vol. 44, pp. 2222–2229, 2008.

[31] D. Silvestre, P. Rosa, J. P. Hespanha, and C. Silvestre, "Stochastic and deterministic fault detection for randomized gossip algorithms," *Automatica*, vol. 78, pp. 46 – 60, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109816305192

[32] R. Motwani and P. Raghavan, *Algorithms and Theory of Computation Handbook*. Chapman & Hall/CRC, 2010, ch. Randomized Algorithms, pp. 12–12. [Online]. Available: http://dl.acm.org/citation.cfm?id=1882757.1882769

[33] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508 – 2530, Jun. 2006.

[34] P. Rosa, C. Silvestre, and M. Athans, "Model falsification using set-valued observers for a class of discrete-time dynamic systems: a coprime factorization approach," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 2928–2942, 2014. [Online]. Available: http://dx.doi.org/10.1002/rnc.3033

[35] J. Shamma and K.-Y. Tu, "Set-valued observers and optimal disturbance rejection," *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 253 –264, feb 1999.

[36] S. Keerthi and E. Gilbert, "Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints," *IEEE Transactions on Automatic Control*, vol. 32, no. 5, pp. 432 – 435, may 1987.

[37] P. Rosa, C. Silvestre, J. Shamma, and M. Athans, "Fault detection and isolation of LTV systems using set-valued observers," *49th IEEE Conference on Decision and Control*, pp. 768–773, December 2010, Atlanta, Georgia, USA.

[38] D. Silvestre, P. Rosa, R. Cunha, J. P. Hespanha, and C. Silvestre, "Gossip average consensus in a byzantine environment using stochastic set-valued observers," in *52nd IEEE Conference on Decision and Control.*, 2013, Florence, Italy.

[39] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal toolbox," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-46, May 2006. [Online]. Available: http://code.google.com/p/ellipsoids

[40] Y. Wei, J. Qiu, and S. Fu, "Mode-dependent nonrational output feedback control for continuous-time semi-markovian jump systems with time-varying delay," *Nonlinear Analysis: Hybrid Systems*, vol. 16, pp. 52 – 71, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1751570X14000600

[41] D. Williams, *Probability with martingales*. Cambridge: Cambridge University Press, 1991.

[42] D. Pollard, *A user's guide to measure theoretic probability*. Cambridge University Press, 2002, vol. 8.