Comparison of Recent Advances in Set-membership Techniques: Application to State Estimation, Fault Detection and Collision Avoidance

Daniel Silvestre[a,b,c]

[a] Department of Electrical and Computer Engineering, NOVA University of Lisbon, Largo da Torre, Caparica, 2829-516, Lisbon, Portugal
[b] COPELABS, Lusofona University, Campo Grande 376, Lisbon, 1749-024, Lisbon, Portugal
[c] Institute for Systems and Robotics, University of Lisbon, Av. Rovisco Pais 1, Lisbon, 1049-001, Lisbon, Portugal

Abstract

There has been a vast body of research on set-membership techniques in recent years. These algorithms compute convex sets that contain the state of a dynamical system given bounds on disturbance and noise signals. Recently, a thorough comparison of zonotopes-based methods against interval arithmetic and ellipsoids has been presented in the literature. However, two main issues were left unexplored: i) added conservatism in the presence of bounds of different kinds such as ℓ_2-norm for the disturbance and an ℓ_∞-norm bound on the noise: ii) set-membership methods can be used in different settings apart from guaranteed state estimation, such as fault detection and isolation and collision avoidance of autonomous vehicles. In this paper, we extend this comparison by considering state estimation, fault detection and isolation, and collision avoidance for interval arithmetic, ellipsoids, zonotopes, constrained zonotopes, polytopes and constrained convex generators in the presence of various combination of bounds for the exogenous signals. The main objective is to compare accuracy, computation time and the scalability of the growth of the data structures required by each set representation. The results indicate that intervals, ellipsoids and zonotopes have a much worse accuracy. The recently introduced Constrained Convex Generators have a negligible increase in computation time in comparison with constrained zonotopes but have a better accuracy when bounds for disturbances, noise and initial conditions are heterogeneous or at least not polytopic.

Preprint submitted to European Journal of Control *October 7, 2024*

Keywords: Observers for Linear Systems, Estimation and fault detection, Guidance, Navigation and Control.

1. Introduction

State estimation of a dynamical system is the problem of producing a value for the unknown state vector, which can be accomplished considering the stochastic and the deterministic settings [11]. The former poses assumptions on the knowledge of the probability distribution of the disturbance and measurement noise whereas the latter consists in providing a description for a set where the state of a dynamical system belongs at some point in time given bounds for the exogeneous signals. In a recent article [5], the authors have compared interval arithmetic, zonotope-based methods and polytopes in their formulation of constrained zonotopes for the problem of guaranteed state estimation. However, two main issues were left unaddressed, namely: i) set-membership techniques can be employed for other interesting problems with different emphasis on what constitutes performance; ii) there have been proposed additional techniques such as A-H polytopes ([29]), Ellipsotopes ([17]), and the more general Constrained Convex Generators (CCGs) [36] that should also be compared.

In this paper, we propose to expand the comparison in [5] to encompass two other applications of interest, namely Fault Detection and Isolation (FDI) and Collision Avoidance. Whereas state estimation requires the observer to compute the set description data structures and possibly a value in its center, FDI or collision avoidance require other operations regarding the sets. In FDI, one needs to check if the set-valued estimates generated by a bank of observers for each faulty scenario are the empty set to invalidate that model. On the other hand, collision avoidance requires testing the intersection of the set-valued estimates with the obstacle. Thus, the further comparisons in this paper should better contextualize how each set description fairs with respect to a) computing the set representation, b) calculating the center of the set, c) checking if a set is empty and d) verifying if two sets intersect. Another overlooked issue with respect to ii) in a fair comparison is how the bounds for the disturbances and noise are selected. If a researcher picks an article proposing an ellipsoid method, most likely the comparison is against other ellipsoid-based algorithms or with the implicit assumption that bounds for disturbances and noise are written using the ℓ_2-norm, and thus directly translated as an ellipsoid and approximated for the competitor's technique. The reverse is also true if the article is about polytopes or constrained zonotopes.

The objective of this paper is to offer a comparison of a variety of set-membership techniques in the three specified application scenarios, namely; i) state estimation, ii) FDI and, iii) collision avoidance of autonomous vehicles. We will be considering examples of linear models for which most of the techniques have been developed. We remark to the reader that as long as the numerical values of all the involved matrices are known up to time instant k, the set-membership task is equivalent irrespective of whether those matrices change over time. Thus, we will be adopting Linear Parameter-Varying (LPV) models introduced by the work of Michael Athans (see [32]) to encompass a class of nonlinear dynamics that can be treated as linear systems when designing observers. These models have widespread applications in aerospace industry, mechatronic systems, automotive, robotic manipulators, vehicle motion, active magnetic bearings, among other academic examples as reported in the survey [15]. We point out that parameters in LPVs are not known at the design phase but can be measured during the execution at a specific time instant. A particular subset of LPVs are the standard Linear Time-Varying (LTV), where the entries are known functions over time and can therefore be determined for each time instant even at design phase.

The approaches for set-valued estimation can also be grouped as *direct* and *indirect* methods. The former, manipulates sets by applying the equivalent set operations to those appearing in the equations that model the dynamics. This means a forward propagation that corresponds to reachability analysis of the dynamics followed by an intersection with the set of state values complying with the measurements [4, 26]. Indirect approaches are based on the idea of running a Luenberger observer to perform a point estimation and then computing the possible set-valued estimate of the errors using forward reachability but avoiding the set intersection. The direct methods have their root on the early work in [30] using ellipsoids and then evolving to parallelotopes in [8]. Later, with the work [33] and [2] it is shown how to perform state estimation using polytopes and zonotopes.

The more recent literature for direct state estimation for LTVs encompasses using interval arithmetic [47, 46, 21], zonotopes [9][22], ellipsoids [18, 7], constrained zonotopes [31], polytopes [42] and even by combining different convex generators [36]. For the sake of completeness, it is also pointed out that these strategies have been extended for nonlinear systems at the expenses of approximating functions and using the same types of set description as for the LTVs as in [1], [2], [16], [25], [48], respectively. The interested reader can get further references in [5] for variations on these methods. The case of considering uncertainties can be accomplished through the convex hull of the sets propagated for each of the vertices of the polytopic dynamics. Such an alternative has been pursued for polytopes stored as vertices in [37] or hyper-planes [12], CCGs in [35, 38] and Constrained Zonotopes in [24].

On the other hand, the indirect methods have gained popularity in recent years. For instance, polytopic and ellipsoidal representations are used in [6]. Using intervals has also been accomplished in [51] and [45] through the use of two sub-observers to compute upper and lower enclosures that can be unified through the convex hull of both sets (which in interval notation is a rather low-complexity method). The application of zonotopes to this approach to state estimation has been accomplished in [10] through the so-called Zonotopic Kalman Filter (ZKF) that selects an observer gain to minimize the FW-radius of the zonotope for the error. Later, the work in [23] compared the indirect methods in [10] both theoretically and numerically.

For nonlinear systems or models with uncertainties, the reduced wrapping effect is an interesting property to avoid that the conservatism is being propagated with the dynamics. Zonotopes have been advocated in the literature since they have been shown to reduce this effect in comparison with ellipsoids and intervals like the work in [49] and [20]. However, zonotopes, ellipsotopes, constrained zonotopes, and any form of polytopes alleviate (or eliminate when all the operations are exact) by increasing the size of the data structures associated with the sets. Therefore, reductions methods must be used to obtain a lower dimensional representation by increasing the hypervolume. Order reductions for zonotopes and constrained zonotopes have been compared in [50]. In order to focus on the data structures themselves, we will avoid order reduction methods or considerations regarding privacy [3] that are outside the scope of the paper.

Regardless of whether we select to represent the error set of an indirect approach or use the set description to directly estimate the state, it is imperative that a thorough comparison is presented that allows the community to better place each of the approaches in terms of computational complexity, accuracy and memory requirements. Moreover, given the aforementioned shortcommings of [5], it is necessary to have such a comparison be performed within the same programming language and paradigm to avoid having drawing conclusions with different implementations that may have been written with distinct applications in mind. Therefore, the main contributions of this paper can be summarized as:

- A comparison is performed for a variety of methods and different combinations of bounds for the disturbances and noise in order to to assess if state estimation is degraded;
- All set representations mentioned in this section have been implemented and are available in `https://github.com/danielmsilvestre/ReachTool`;
- The problem of FDI is considered from the model falsification point-of-view showcasing performance to check for emptiness of the produced sets and possible certificates;
- The use case of collision avoidance with convex obstacles is also investigated to assess performance of solving a feasibility problem with constraints given by the set-membership techniques.

The remainder of the paper is organized as follows. In Section 3.1, we formalize the type of model assumed for the comparisons in this paper and present the set operations definitions in the various proposals for state estimation in LPV models. Sections 3.2 and 3.3 expand the details of the applicability of set-membership techniques to fault detection and isolation, and, the problem of collision avoidance in autonomous vehicles. Simulations for the various scenarios are presented in Section 4, while conclusions and directions of future work are offered in Section 5.

Notation: In this paper, we denote by $\mathbf{v}$ an anonymous variable in an optimization problem that corresponds to a possible value for the vector v. The Minkowski sum of two sets X and Y is defined as $X \oplus Y := \{v + u : v \in X, u \in Y\}$. The infinity norm of a vector is denoted by $\|v\|_\infty$ and corresponds to $\max_i |v_i|$ for the absolute value function $|a|$ for the scalar a. The vector of all ones, all zeros and the identity matrix is denoted by $\mathtt{1}_n$, $\mathtt{0}_n$ and I_n for dimension n. If we require non-square matrices, dimensions will be updated as $n \times r$ for n rows and n columns. When the modulus is applied to a matrix it is meant element-wise, i.e., $|A| = [|a_{ij}|], \forall i, j$. The Moore-Penrose inverse of a matrix A is denoted by $A^\dagger$. The function $\mathrm{diag}(v)$ returns a diagonal matrix whose diagonal elements consists of the entries in vector v.

2. Set Representations

Before introducing the applications related to set-membership techniques, we provide a presentation of the set representations under the same format and present the three required set operations when dealing with linear dynamics without uncertainties:linear map, Minkowski sum and intersection through a map. In particular, we will highlight the reasons why some of the operations can be performed in closed-form while the others require iterative procedures or the solution of optimization problems.

2.1. Interval Arithmetic

The simplest type of sets would be intervals, which translate into hyper-rectangles in $\mathbb{R}^n$. In order to store them, there are two viable representation: i) using the minimum and maximum value for each coordinate like $x(k) \in [a, b]$, where $a, b \in \mathbb{R}^n$; or by saving the midpoint and radius like $x(k) \in \begin{bmatrix} m - r, & m + r \end{bmatrix}$ with $m, r \in \mathbb{R}^n$. The latter formulation is more amenable to the computations required for the state estimation.

Definition 1 (Interval). *A set I is an interval defined by the tuple $(m, r) \in \mathbb{R}^n \times \mathbb{R}^n$ such that:*

$$I = \{\xi : m - r \leq \xi \leq m + r\},$$

where the above inequality applies element-wise.

Having Definition 1, one can introduce the three required set operations for intervals as:

Lemma 2 (Interval operations). *Consider three intervals as in Definition 1:*

- $Z = (m_z, r_z) \subset \mathbb{R}^n$;
- $W = (m_w, r_w) \subset \mathbb{R}^n$;
- $Y = (m_y, r_y) \subset \mathbb{R}^m$;

and a matrix $R \in \mathbb{R}^{m \times n}$ and a vector $t \in \mathbb{R}^m$. The three set operations are defined as:

$$RZ + t \subseteq (Rm_z + t, |R|r_z)$$
$$Z \oplus W = (m_z + m_w, r_z + r_w)$$

$$Z \cap_R Y \subseteq Z \cap \left(\bigcap_{i=1}^{m} \tilde{Y}_i \right) \tag{1}$$

where the interval for each observation $\tilde{Y}_i = (\tilde{m}_i, \tilde{r}_i)$ *is defined as*

$$\tilde{m}_i = \operatorname{diag}(R_{i,:})^{-1} \left[m_y \mathbb{1}_n - \left(\mathbb{1}_n R_{i,:} - \operatorname{diag}(R_{i,:}) \right) m_z \right]$$
$$\tilde{r}_i = \left| \operatorname{diag}(R_{i,:})^{-1} \left[r_y \mathbb{1}_n + \left(\mathbb{1}_n R_{i,:} - \operatorname{diag}(R_{i,:}) \right) r_z \right] \right|$$

for a non-zero i*-th row* $R_{i,:}$ *and the standard intersection being given by:*

$$Z \cap W = \left(\frac{\min(b_z, b_w) + \max(a_z, a_w)}{2}, \frac{\min(b_z, b_w) - \max(a_z, a_w)}{2} \right)$$

using

$$a_z = m_z - r_z,\ b_z = m_z + r_z,\ a_w = m_w - r_w,\ b_w = m_w + r_w.$$

Proof. Both the linear map and the Minkowski sum expressions can be found in [46] where the expression for the linear map can be seen as taking the box of the zonotope resulting from the linear map.

Lastly, the generalized intersection operation can be split by considering the interval $\tilde{Y}_i$ obtained by each measurement represented by the i-th row $R_{i,:}$ of matrix R. This means that after computing $\tilde{Y}_i$, these sets need to be intersected with Z as written in (1). A point resulting from the generalized intersection must satisfy being a member of both Z and Y after multiplying by R, which results in the equation

$$R_{i,:}(m_z + \operatorname{diag}(\Delta_z) r_z) = m_y + \Delta_y r_y, \|\Delta_z\|_\infty \leq 1, |\Delta_y| \leq 1, \Delta_z \in \mathbb{R}^n, \Delta_y \in \mathbb{R}.$$

From the definition of midpoint of an interval, this corresponds to the average of the minimum and maximum values along any coordinates. Consider that the maximum (conversely the minimum) of each coordinate corresponds to setting the remaining to be their minimum (conversely maximum), while still satisfying the equation. Therefore, the terms associated with the radius average to zero and we can stack each of the n equations where we replaced all of the remaining variables by the average of the minimum and maximum (i.e., the midpoint m_z):

$$\begin{aligned} & \operatorname{diag}(R_{i,:}) \tilde{m}_i + \left(\mathbb{1}_n R_{i,:} - \operatorname{diag}(R_{i,:}) \right) m_z = \mathbb{1}_n m_y \\ \iff & \operatorname{diag}(R_{i,:}) \tilde{m}_i = \mathbb{1}_n m_y - \left(\mathbb{1}_n R_{i,:} - \operatorname{diag}(R_{i,:}) \right) m_z \\ \iff & \tilde{m}_i = \operatorname{diag}(R_{i,:})^{-1} \left[\mathbb{1}_n m_y - \left(\mathbb{1}_n R_{i,:} - \operatorname{diag}(R_{i,:}) \right) m_z \right], \end{aligned}$$

which is well defined for non-zero rows of R.

The definition of radius is the average of the difference between the maximum and the minimum, or equivalently, the absolute value between the two extreme points. Similarly to the midpoint computation, we need to stack n equations where in each we allow a coordinate to change and fix the remaining from the interval Z. We will denote $\tilde{p}$ as the stack of the changing coordinates and p as the remainder of applying $R_{i,:}$ to a point in Z, leading to the stack of equations:

$$\begin{aligned} & \tilde{p} + p = \mathbb{1}_n m_y + \Delta_y \mathbb{1}_n r_y, \|\Delta_y\|_\infty \leq 1, \Delta_y \in \mathbb{R}^n \\ \iff & \tilde{p} = \mathbb{1}_n m_y + \Delta_y \mathbb{1}_n r_y - p, \|\Delta_y\|_\infty \leq 1, \Delta_y \in \mathbb{R}^n. \end{aligned} \tag{2}$$

The next step is to compute the radius of the signal on both sides of the equation:

$$\begin{aligned} & |\mathrm{diag}(R_{i,:})\tilde{r}_i| = \frac{1}{2}\left| \underbrace{(\mathbb{1}_n m_y + \mathbb{1}_n r_y - \min p)}_{\text{max of right-hand side of (2)}} - \underbrace{(\mathbb{1}_n m_y - \mathbb{1}_n r_y - \max p)}_{\text{min of right-hand side (2)}} \right| \\ \iff & |\mathrm{diag}(R_{i,:})\tilde{r}_i| = |\mathbb{1}_n r_y + 0.5\,(\max p - \min p)| \\ \iff & |\mathrm{diag}(R_{i,:})\tilde{r}_i| = |\mathbb{1}_n r_y + (\mathbb{1}_n R_{i,:} - \mathrm{diag}(R_{i,:}))\, r_z| \\ \iff & \tilde{r}_i = \left|\mathrm{diag}(R_{i,:})^{-1}\right| |\mathbb{1}_n r_y + (\mathbb{1}_n R_{i,:} - \mathrm{diag}(R_{i,:}))\, r_z| \\ \iff & \tilde{r}_i = \left|\mathrm{diag}(R_{i,:})^{-1} \left[\mathbb{1}_n r_y + (\mathbb{1}_n R_{i,:} - \mathrm{diag}(R_{i,:}))\, r_z\right]\right|, \end{aligned}$$

which concludes the proof. ■

One of the main questions that is often overlooked in this type of comparison is how to obtain an actual estimate to be used by a controller. When using intervals, the estimate can simply be the midpoint of the interval, meaning a very efficient computation.

2.2. Ellipsoids

Given the natural occurrence of ℓ_2-norm bounds related to the magnitude of a physical noise signal, a natural approach is to consider ellipsoids. In order to do so, one needs a center c and a shape matrix Q as in the following definition.

Definition 3 (Ellipsoids). *A set E is an ellipsoid defined by the tuple $(c, Q) \in \mathbb{R}^n \times \mathbb{R}^{n\times n}$ such that:*

$$E = \{\xi : (\xi - c)^\intercal Q^{-1}(\xi - c) \leq 1\}.$$

In this formulation, there is no need to store an extra vector for the right hand-side of the above inequality. Let us introduce function regularize applied to an ellipsoid $Z = (c, Q) \in \mathbb{R}^n$ where Q has rank r to be:

$$\text{regularize}(Z) = (c, 0.5(Q_r + Q_r^\intercal))$$

where

$$Q_r = Q + U \begin{bmatrix} \mathsf{0}_{r\times r} & \mathsf{0}_{r\times(n-r)} \\ \mathsf{0}_{r\times(n-r)} & \text{cnst} I_{n-r} \end{bmatrix} U^\intercal$$

with cnst being a large constant.

Having both definitions, one can introduce the three required set operations as:

Lemma 4 (Ellipsoids operations). *Consider three ellipsoids as in Definition 3:*

- $Z = (c_z, Q_z) \subset \mathbb{R}^n$;
- $W = (c_w, Q_w) \subset \mathbb{R}^n$;
- $Y = (c_y, Q_y) \subset \mathbb{R}^m$;

and a matrix $R \in \mathbb{R}^{m\times n}$ and a vector $t \in \mathbb{R}^m$. The three set operations are defined as:

$$RZ + t = (Rc_z + t, RQ_zR^\intercal)$$
$$Z \oplus W \subseteq \left(c_z + c_w, \left(1 + \frac{1}{\beta^\star}\right) Q_z + (1 + \beta^\star)Q_w\right)$$

with $\beta^\star$ as the steady-state solution of the iteration:

$$\beta_{k+1} = \left(\frac{\sum_{i=1}^n \frac{1}{1+\beta_k\lambda_i}}{\sum_{i=1}^n \frac{\lambda_i}{1+\beta_k\lambda_i}}\right)^{\frac{1}{2}}$$

and λ_i as the eigenvalues of the matrix $Q_z^{-1}Q_w$.

$$Z \cap_R Y \subseteq \left(c^+, \left(E^+\right)^{-1}\right),$$

where

$$E^+ = \gamma^\star E_z + (1 - \gamma^\star)\tilde{E}_y$$
$$c^+ = \left(E^+\right)^{-1} \left(\gamma^\star E_z c_z + (1 - \gamma^\star)\tilde{E}_y c_y\right)$$

for, $E_z = Q_z^{-1}$*,* $E_y = \left(\text{regularize}\left(\tilde{Q}_y\right)\right)^{-1}$*,* $\left(c_y, \tilde{Q}_y,\right) = R^\dagger Y$ *and* $\gamma^\star$ *being the value of* γ *satisfying*

$$\sum_{i=1}^{n} \frac{1-\lambda_i}{\gamma + (1-\gamma)\lambda_i} = 0$$

using λ_i *to denote the eigenvales of* $E_z^{-1}E_y$*.*

Proof. The Minkowski expression proof can be found in [13] whereas the remaining are given in [19] ∎

The center of the ellipsoid can be used as the state estimate given that it is the center of the convex set.

2.3. Zonotopes

The main disadvantage in terms of accuracy from the intervals is that it does not allow for facets that are not aligned with the axis. The first step is considering zonotopes that are characterized by having all its facets be symmetric in pairs with relation to the center of the set. One possible way is the following definition.

Definition 5 (Zonotopes [2]). *A set* Z *is a zonotope defined by the tuple* $(H, p) \in \mathbb{R}^{n\times n_g} \times \mathbb{R}^n$ *such that:*

$$Z = \{H\xi + p : \|\xi\|_\infty \leq 1\}.$$

We remark that zonotopes can be viewed as a linear transformation of the ℓ_∞ unit ball in the same way that ellipsoids can be viewed as a linear transformation of the ℓ_2 unit ball. This will be quite helpful in the intuition behind constrained convex generators that unites all the set-membership approaches. Let us also define the function box(Z) for $Z \in \mathbb{R}^n$ with n_g generators that returns a zonotope $Z_b = (H_b, p_b)$ that is also a box:

$$H_b = \text{diag}\left(|H|1_{n_g}\right), p_b = p.$$

Having Definition 5, the needed set operations are defined as:

Lemma 6 (Zonotope operations [2]). *Consider three zonotopes as in Definition 5:*

- $Z = (H_z, p_z) \subset \mathbb{R}^n$*;*

- $W = (H_w, p_w) \subset \mathbb{R}^n$;
- $Y = (H_y, p_y) \subset \mathbb{R}^m$;

and a matrix $R \in \mathbb{R}^{m\times n}$ and a vector $t \in \mathbb{R}^m$. The three set operations are defined as:

$$
\begin{aligned}
RZ + t &= (RH_z, Rp_z + t)\\
Z \oplus W &= \left(\begin{bmatrix} H_z & H_w \end{bmatrix}, p_z + p_w\right)\\
Z \cap_R Y &\subseteq \left(\hat{H}_m, \hat{p}_m\right),
\end{aligned}
$$

where $\left(\hat{H}_m, \hat{p}_m\right) = Z \cap_R \mathrm{box}(Y)$, with $\mathrm{box}(Y) = (\mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_m), d)$. Then, the intersection can be found from the following relationship with $\hat{H}_0 = H_z$ and $\hat{p}_0 = p_z$:

$$
\begin{aligned}
\hat{H}_i &= \begin{bmatrix} (I_n - \lambda_i r_i^\intercal)\hat{H}_{i-1} & \sigma_i \lambda_i \end{bmatrix}\\
\hat{p}_i &= \hat{p}_{i-1} + \lambda_i (d_i - r_i^\intercal \hat{p}_{i-1})\\
\lambda_i &= \frac{\hat{H}_{i-1}\hat{H}_{i-1}^\intercal r_i}{r_i^\intercal \hat{H}_{i-1}\hat{H}_{i-1}^\intercal r_i + \sigma_i^2}
\end{aligned}
$$

where $R = \begin{bmatrix} r_1^\intercal \\ \vdots \\ r_m^\intercal \end{bmatrix}$.

Proof. The proof can be found in [2]. ∎

The center of the zonotope can still be used as the estimate for the state of the dynamical system since it still represents the center of the set, following the analogy between ellipsoids and zonotopes.

2.4. Constrained Zonotopes

An improvement on zonotope is to consider additional linear constraints on the set to allow to represent general polytopes [31]. The formal definition for a constrained zonotope is given as:

Definition 7 (Constrained Zonotope [31]). *A set Z is a constrained zonotope defined by the tuple $(G, c, A, b) \in \mathbb{R}^{n\times n_g} \times \mathbb{R}^n \times \mathbb{R}^{n_c\times n_g} \times \mathbb{R}^{n_c}$ such that:*

$$Z = \{G\xi + c : \|\xi\|_\infty \le 1, A\xi = b\}.$$

The main difference in comparison with the standard zonotopes is the inclusion of the equality constraint $Ax = b$. Such change also means that all the three set operations can be performed in closed-form.

Lemma 8 (Set operations [31]). *Consider three constrained zonotopes as in Definition 7:*

- $Z = (G_z, c_z, A_z, b_z) \subset \mathbb{R}^n$;
- $W = (G_w, c_w, A_w, b_w) \subset \mathbb{R}^n$;
- $Y = (G_y, c_y, A_y, b_y) \subset \mathbb{R}^m$;

and a matrix $R \in \mathbb{R}^{m\times n}$ and a vector $t \in \mathbb{R}^m$. The three set operations are defined as:

$$RZ + t = (RG_z, Rc_z + t, A_z, b_z)$$

$$Z \oplus W = \left(\begin{bmatrix} G_z & G_w \end{bmatrix}, c_z + c_w, \begin{bmatrix} A_z & 0 \\ 0 & A_w \end{bmatrix}, \begin{bmatrix} b_z \\ b_w \end{bmatrix} \right)$$

$$Z \cap_R Y = \left(\begin{bmatrix} G_z & 0 \end{bmatrix}, c_z, \begin{bmatrix} A_z & 0 \\ 0 & A_y \\ RG_z & -G_y \end{bmatrix}, \begin{bmatrix} b_z \\ b_y \\ c_y - Rc_z \end{bmatrix} \right).$$

Proof. The proof can be found in [31]. ■

By including the equality, it allows for the definition of the intersection at the expenses of adding additional generator variables that are constrained to belong to the intersection. Given that the sets are no longer symmetric, a lot of possibilities can be taken as the estimate for the state. A very efficient alternative is to find the minimum norm solution to the equality $Ax = b$ and then use that as the numerical values for the generators.

2.5. Polytopes

The previous formulation represents general polytopes that were typically stored in a different formulation. For the sake of completeness, we have also included the standard format to store polytopes.

Definition 9 (Polytopes). *A set P is a polytope defined by the tuple $(A, b, C, d, n) \in \mathbb{R}^{n_{c1}\times n_g} \times \mathbb{R}^{n_{c1}} \times \mathbb{R}^{n_{c2}\times n_g} \times \mathbb{R}^{n_{c2}}$ such that:*

$$Z = \{\xi : A\xi \leq b, C\xi = d\}.$$

Notice that we must store the size of the space since there might be additional generator variables used as auxiliary terms. Also, the equality constraint could be represented as two inequalities as an alternative. The three operations are still performed in closed-form but with a key difference. In the current formulation, intersections are quite easy to define but the dynamics matrix must be non-singular, whereas constrained zonotopes can allow for a general linear map but add more generator variables for the intersection. There are alternatives to deal with this issue such as in [42, 40, 41, 43]. Both strategies are implemented, although for the physical systems being tested, the dynamics matrix is always going to be non-singular. In the following definition, we will be using a Matlab-like notation where $A[:, i : j]$ stands for all the rows of matrix A (: is used to denote all rows) and columns from i to j.

Lemma 10 (Set operations). *Consider three polytopes as in Definition 9:*

- $Z = (A_z, b_z, C_z, d_z, n) \subset \mathbb{R}^n$*;*
- $W = (A_w, b_w, C_w, d_w, n) \subset \mathbb{R}^n$*;*
- $Y = (A_y, b_y, C_y, d_y, m) \subset \mathbb{R}^m$*;*

and a non-singular matrix $R \in \mathbb{R}^{m\times n}$ *and a vector* $t \in \mathbb{R}^m$*. The three set operations are defined as:*

$$
\begin{aligned}
RZ + t = \Big(& \begin{bmatrix} A_z[:, 1 : n]R^{-1} & A_z[:, (n + 1) : n_g] \end{bmatrix}, \\
& b_z + A_z[:, 1 : n]R^{-1}t, \\
& \begin{bmatrix} C_z[:, 1 : n]R^{-1} & C_z[:, (n + 1) : n_g] \end{bmatrix}, \\
& d_z + C_z[:, 1 : n]R^{-1}t, m \Big)
\end{aligned}
$$

$$
Z \oplus W = \left(\begin{bmatrix} A_z & 0 \\ 0 & A_w \end{bmatrix}, \begin{bmatrix} b_z \\ b_w \end{bmatrix}, \begin{bmatrix} I_n & \begin{array}{cccc} -I_n & 0 & -I_n & 0 \end{array} \\ 0 & \begin{array}{|cc} \hline C_z & 0 \\ 0 & C_w \end{array} \end{bmatrix}, \begin{bmatrix} 0 \\ d_z \\ d_w \end{bmatrix}, n \right)
$$

$$
Z \cap_R Y = \left(\begin{bmatrix} A_z & 0 \\ 0 & A_y \end{bmatrix}, \begin{bmatrix} b_z \\ b_y \end{bmatrix}, \begin{bmatrix} \begin{array}{cc} C_z & 0 \\ 0 & C_y \end{array} \\ \hline \begin{array}{cccc} R & 0 & -I_n & 0 \end{array} \end{bmatrix}, \begin{bmatrix} d_z \\ d_y \\ 0 \end{bmatrix}, n \right).
$$

Proof. The derivation is equivalent to that in [42]. ∎

We remark to the reader that all operations have closed-form expressions given that the use of equality constraints allow to represent intersection through the addition of auxiliary variables, much like was done for the case of constrained zonotopes.

2.6. Constrained Convex Generators

From the presented intuition related to the surveyed set representations, it becomes apparent that ellipsoids are linear transformations of the ℓ_2 unit ball as zonotopes are linear maps of the ℓ_∞ unit ball whereas constrained zonotopes added an equality constraint to allow representation of the intersection operation in closed form. Similarly, polytopes can be viewed as a convex set represented by $Ax \leq b$ to which an equality constraint can directly be represented using two inequalities or by explicitly storing them in a different data structure as was done in Section 2.5. Nevertheless, the key components for a set-membership technique is the possibility to store simple convex sets along with an equality constraint to ensure that the intersection has closed-form expression. This is precisely the definition of constrained convex generators that are given in a formal definition by:

Definition 11 (Constrained Convex Generators [36]). *A Constrained Convex Generator (CCG) $\mathcal{Z} \subset \mathbb{R}^n$ is defined by the tuple $(G, c, A, b, \mathfrak{C})$ with $G \in \mathbb{R}^{n \times n_g}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{n_c \times n_g}$, $b \in \mathbb{R}^{n_c}$, and $\mathfrak{C} := \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_{n_p}\}$ such that:*

$$\mathcal{Z} = \{G\xi + c : A\xi = b, \xi \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_{n_p}\}.$$

Having the above definition, the set operations can still be formulated as:

Lemma 12 (Set operations [36]). *Consider three Constrained Convex Generators (CCGs) as in Definition 11:*

- $Z = (G_z, c_z, A_z, b_z, \mathfrak{C}_z) \subset \mathbb{R}^n$;
- $W = (G_w, c_w, A_w, b_w, \mathfrak{C}_w) \subset \mathbb{R}^n$;
- $Y = (G_y, c_y, A_y, b_y, \mathfrak{C}_y) \subset \mathbb{R}^m$;

and a matrix $R \in \mathbb{R}^{m \times n}$ and a vector $t \in \mathbb{R}^m$. The three set operations are defined as:

$$RZ + t = (RG_z, Rc_z + t, A_z, b_z, \mathfrak{C}_z)$$

$$Z \oplus W = \left(\begin{bmatrix} G_z & G_w \end{bmatrix}, c_z + c_w, \begin{bmatrix} A_z & 0 \\ 0 & A_w \end{bmatrix}, \begin{bmatrix} b_z \\ b_w \end{bmatrix}, \{\mathfrak{C}_z, \mathfrak{C}_w\} \right)$$

$$Z \cap_R Y = \left(\begin{bmatrix} G_z & 0 \end{bmatrix}, c_z, \begin{bmatrix} A_z & 0 \\ 0 & A_y \\ RG_z & -G_y \end{bmatrix}, \begin{bmatrix} b_z \\ b_y \\ c_y - Rc_z \end{bmatrix}, \{\mathfrak{C}_z, \mathfrak{C}_y\} \right).$$

Proof. The derivation can be found in [36]. ■

We remark that all of the previously introduced set representations are specific instances of constrained convex generator and that are additional sets that can be represented under this formulation. In Figure 1, it is depicted the intersection of a polytope and an ellipsoid and the intersection of two ellipsoids that can be both modeled as constrained convex generator but cannot be represented neither using constrained zonotopes nor ellipsoids. In particular, we have that:

- an interval corresponds to $(G, c, [\,], [\,], \|\xi\|_\infty \leq 1)$, for a diagonal matrix G;
- a zonotope is given by $(G, c, [\,], [\,], \|\xi\|_\infty \leq 1)$;
- an ellipsoid is defined by $(G, c, [\,], [\,], \|\xi\|_2 \leq 1)$, for a square matrix G;
- a CZ or polytope is $(G, c, A, b, \|\xi\|_\infty \leq 1)$;
- a convex cone in $\mathbb{R}^n$ is $(G, c, [\,], [\,], \xi \geq 0)$;
- ellipsotopes are given by $(G, c, A, b, \|\xi\|_{p_1} \leq 1, \cdots, \|\xi\|_{p_m} \leq 1)$, for some $p_i > 0$, $1 \leq i \leq m$;
- AH-polytopes are given by $(G, c, [\,], [\,], A\xi \leq b)$.

3. Application of Set-membership Techniques

In this paper, we use 3 particular applications of set-membership techniques, although these are quite representative of other applications, since we can pinpoint the main operation procedure that is required in each case. We wanted to showcase applications where, aside from building the set data structures, the main operations can be grouped as: i) *finding the center of*

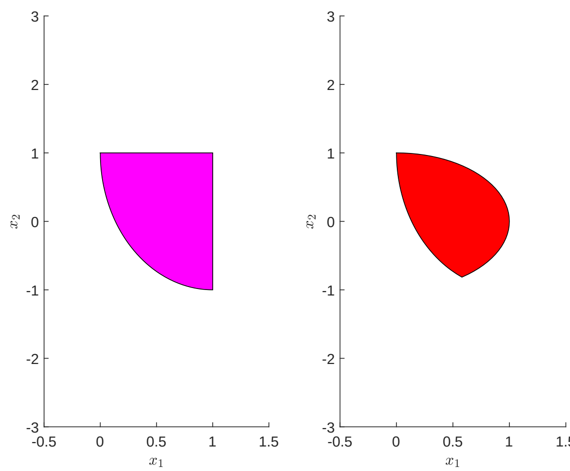

Figure 1: Two sets that can be modeled using constrained convex generators. On the left: set resulting from the intersection of a square with an ellipse. On the right: intersection of two ellipses.

the set or its boundary; ii) *checking whether a set is empty*; and iii) *testing for the intersection of sets.* State estimation for dynamical systems belongs to class i). However, we could have provided other instances where the same issue appears like constructing constraints for Model Predictive Control to ensure feasibility of the optimization problem, calculating Robust Positively Invariant Sets, estimating the uncertainty of parameter estimation in a consensus algorithm, among many others. In all such cases, the problem reduces to building a set that represents all possible values for a given uncertain variable and then using that description to either calculate a center point, introduce in optimization problems or provide a measure of its size. The standard version of i) is state estimation for dynamical systems as a deterministic version of what is accomplished with stochastic observers like the Kalman filter.

Class ii) translates problems of testing whether the measurements are compliant with a given model and bounds for the exogeneous signals. Instances can arise in Fault Detection and Isolation (our selected application) since we are checking if each model with the presence of a different fault is valid, but we could have picked other examples. In Multiple Model Control, set-membership techniques can be used to check which model is currently compatible with the measurements whereas in the Distinguishability problem we would be interested in testing how many measurements are required before we are able to separate a given set of models regardless of initial conditions or noise and disturbances. Given that an attacker can be viewed as a malicious fault design, FDI also plays a key role in detecting the presence of intruders in linear iterative distributed algorithms.

Lastly, since the produced sets by deterministic methods in Section 2 represent all possible trajectories for a system, intersecting with other sets can be used to test for safety. In particular, if those sets represent obstacles, set-membership can be used for collision avoidance by testing the intersection of the set describing all possible values for future time instants with the obstacles. However, this can be used for active fault detection by computing actuation signals that render an empty intersection of the sets for the faulty and non-faulty models. Regardless of the specifics of the application, testing intersections typically requires solving an optimization problem. Thus, class iii) is used to assess this operation and how the set representations fair against each other when intersections need to be computed.

3.1. State Estimation using Set-membership approaches

In this paper, the problem of state estimation for LPVs using a set-membership approach consists in finding a set of possible values given the dynamics and measurements obtained from the system modeled as:

$$\begin{aligned} x(k+1) &= A(\rho(k))x(k) + B(\rho(k))u(k) + L(\rho(k))d(k) \\ y(k) &= C(\rho(k))x(k) + N(\rho(k))w(k) \end{aligned} \tag{3}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^{n_u}$, $d(k) \in \mathbb{R}^{n_d}$, $y(k) \in \mathbb{R}^m$ and $w(k) \in \mathbb{R}^{n_w}$ are the system state, input, disturbance signal, output and noise, respectively. The parameter $\rho(k)$ can be measured at time k, which allows to model some non-linear systems as (3) while not posing additional difficulties for the estimation using a set-membership approach. To lighten the notation, we will consider $A_k := A(\rho(k))$ and similarly for all the remaining matrices in (3). Moreover, in order to have a well-posed problem, we assume that all unknown signals are bounded within a compact set denoted by the correspondent capital letter, i.e., $x(0) \in X(0)$, $d(k) \in D(k)$ and $w(k) \in W(k)$. The only exception to this assumption would be the use of CCGs that can accommodate, for instance, cones as the generator sets.

The state estimation problem can be summarized as:

Problem 13. *Given compact sets* $X(0)$*,* $D(k)$ *and* $W(k)$ *for all* $k \geq 0$ *and measurements* $y(k)$*, how to compute a set* $X(k)$ *such that it is guaranteed that* $x(k) \in X(k)$*,* $\forall k \geq 0$*.*

Problem 13 can be solved iteratively from the previous estimate by first performing the propagate phase corresponding to the set-valued version of the dynamics equation:

$$X_{prop}(k+1) = A_k X(k) \oplus B_k u(k) \oplus L_k D(k)$$

where a matrix multiplying a set corresponds to applying that linear map to all vectors in the set. In a similar fashion, given the measurement $y(k)$, the update phase to conform with the information $y(k)$ corresponding to the set-valued version of the measurement equation can be performed by:

$$X(k) = X_{prop}(k) \cap_{C_k} y(k) \oplus N_k W(k)$$

where the symbol $\cap_{C_k}$ stands for the intersection through the map C_k such that both sets being intersected constrain the possible values of $x(k)$.

3.2. Fault Detection and Isolation

An algorithm that produces set-valued estimates possessing the property that $x(k) \in X(k)$, $\forall k$ can be used in the logic to perform fault detection and isolation. Intuitively, if the algorithm does not introduce conservatism, $X(k)$ is the set of all possible valid solutions $x(k)$ that satisfy the set of equations in (3) and verifies all past measurements $y(0), \cdots, y(k)$. With that view present, fault detection corresponds to checking whether $X(k)$ is empty, which would invalidate the assumed model for the system (please view examples of this usage in [31, 41] for polytopes and constrained zonotopes). Fault isolation corresponds to extending this concept to that of distinguishing which model is actually generating the measurement data. The concept is also known as distinguishability [44] in the literature. When isolating a fault, a bank of observers has to be computed using the set-membership approach for each model assuming the specific combination of faults that are possible to happen.

Let us construct a concrete example considering two faults for the system in (3) where: i) there is an unmodeled disturbance vector f_1; and, ii) there is additional noise f_2 in the measurements. Moreover, if no assumption of exclusivity among faults is posed, FDI requires 4 models, namely M_0 corresponding to (3) (the nominal model), and three additional ones:

$$M_1 : \begin{cases} x(k+1) = A_k x(k) + B_k u(k) + L_k d(k) + f_1 \\ y(k) = C_k x(k) + N_k w(k) \end{cases}$$

$$M_2 : \begin{cases} x(k+1) = A_k x(k) + B_k u(k) + L_k d(k) \\ y(k) = C_k x(k) + N_k w(k) + f_2 \end{cases}$$

$$M_{12} : \begin{cases} x(k+1) = A_k x(k) + B_k u(k) + L_k d(k) + f_1 \\ y(k) = C_k x(k) + N_k w(k) + f_2 \end{cases}.$$

Let us associate $X_i(k)$ to the set produced by a set-membership algorithm for the model labeled as i. Then, the logic for FDI is as follows:

- $X_0(k) = \emptyset$ means that a fault is detected;
- $X_0(k) = \emptyset$ and $\forall i \neq j, X_i(k) = \emptyset$, then fault j is isolated and is guaranteed to be the one occurring.

We remark to the reader that set-membership FDI has the advantage of providing guarantees (meaning that once it declares a fault, it must be happening) but has a combinatorial growth in the number of filters since a model has to be computed for each possible valid scenario.

3.3. Collision Avoidance for Autonomous Vehicles

Multi-agent missions for surveillance or exploration have seen the introduction of distributed iterative algorithms with the use of set-membership techniques to avoid collisions with obstacles or other nodes by computing the set of all possible positions for both the agent and its surroundings [27, 28]. In essence, given that a set $X(k)$ translates all valid instances for $x(k)$, it is possible to detect potential collisions by constructing a set-valued estimate for the position of the vehicle and all remaining obstacles. Notice that using a linear map and a projection matrix, one can retrieve the set-valued estimate for the position from the whole $X(k)$, although it is more efficient to construct a dynamical model only for the position variables.

Two main steps influence the overall performance of the application of set-membership techniques to collision avoidance: i) the set operations themselves, ii) the computation time to construct and solve the feasibility problem to check whether the vehicle position set intersects with that of an obstacle. Step ii) is worsen since to avoid non-convex problems, it is required to cycle over each obstacle in the list.

4. Simulation

In order to assess the characteristics of all set-membership techniques in the surveyed use cases, a thorough simulation was conducted and the code is made available at GitHub so that the research community can have a fair comparison of all the methods implemented in Matlab and resorting to the same optimization solver. Simulations were run in Matlab R2018a running on a HP machine with a Intel Core i7-8550U CPU @ 1.80GHz and 12 GB of memory resorting to Yalmip as the language to model optimization problems and Mosek as the underlying solver. Videos, figures and code can be found in `https://github.com/danielmsilvestre/SETcomparison`.

4.1. Comparison for State Estimation

An important aspect of set-membership algorithms applied to state estimation is that they are going to be applied to physical systems. In such cases, matrix A_k is invertible for continuous-time systems that were discretized via Euler method. For singular A_k matrices, structures like polytopes or ellipsoids have particular bad performance either due to numerical issues or due to a larger increase in the number of auxiliary variables required for the set representation. Therefore, some of the conclusions in [5] need to be complemented by the comparison in this section.

We have selected the physical systems from the tutorials in `https://ctms.engin.umich.edu/CTMS/index.php`, namely: i) *cruise control*, ii) *motor speed*, iii) *suspension*, iv) *inverted pendulum* and v) *aircraft pitch*. All parameters for the models and their correspondent dynamical equations can be found in the mentioned link. The sampling time Ts = 0.1 s and a state feedback controller was designed to each model with eigenvalues for the closed-loop dynamics matrix that range from 0.01 to 0.1 with a linear spacing among them. Simulations are run for 20 time instants as it is sufficient to highlight the trends for the reported 4 quantities relevant in assessing each of the algorithms, namely: i) estimation error, ii) computation time to produce both the set-valued estimates and the vector estimate, iii) hypervolume of the sets, and iv) number of double values necessary to represent the data structures for each set-membership approach.

The operations to produce a vector estimate from the sets $X(k)$ and to compute the hyper-volume are not universal throughout the literature. For intervals, ellipsoids and zonotopes, since all sets have an inherent symmetry with respect to a center point, the vector estimate was assumed to be the center of each set. Constrained zonotopes and CCGs share the idea of an auxiliary set (defined by means of a linear equation) that is deformed by a linear transformation to obtain $X(k)$. For that reason, the vector estimate corresponds to the linear transformation applied to the minimum ℓ_2-norm solution of the linear equation. On the other hand, polytopes are stored in a different formulation and was selected the Chebyshev center that can be formulated as a linear program.

Intervals, zonotopes and ellipsoids all have closed-form expressions for their hyper-volume that can be found in the surveyed literature regarding these methods. However, the same does not hold for polytopes, constrained zonotopes and CCGs. These last were approximated by a ray shooting technique that consists in selecting a center point (the vector estimate) followed by random rays shooting until hitting the surface of the convex body. Those points are taken as vertices of an inner polytope and the approximation for the volume uses a triangulation algorithm that is available in Matlab exchange and using qhull. The procedure is stopped when the hyper-volume increase is below 1% from the previous iteration after taking 800 new vertices. In order to improve the performance, in the initial iterations, the vertices are taken as the canonical vectors (get an inner hyper-rectangle) and then for all vertices of a unit hyper-cube (to get an inner diamond-shape or rhombic approximation).

For the *cruise* control model, we depict in Figure 2 the evolution of computation time to obtain $X(k)$ and a vector estimate. Given the scalar state for this model, algorithms are very close in terms of hyper-volume and estimation error and the space for the data structures trend is similar to the remaining examples in this section. As observed in Figure 2, zonotopes were the fastest with an average computation time for iteration of 5.23×10^{-5} s, followed by ellipsoids, CCGs and constrained zonotopes with similar performance among them (5.94×10^{-4} s, 3×10^{-4} s and 2.64×10^{-4} s, respectively). Intervals and polytopes behave similarly with 0.25 s and 0.27 s on average. These values shall be compared with the models with larger state space size.

An interesting trend can be observed from the *motor speed* model with respect to the hyper-volume of the produced set-valued estimates. Intervals have the worst accuracy with a volume of 4.85 at iteration 20 followed by

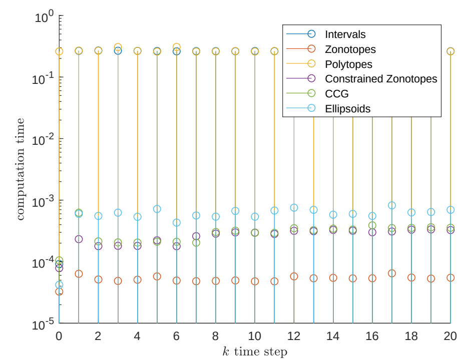

Figure 2: Computation time to obtain $X(k)$ and a vector estimate for the cruise control model when using a unit ℓ_∞-ball as bound for the disturbance and a unit ℓ_2-ball for the noise.

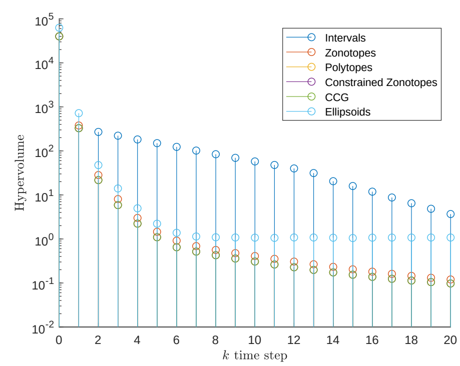

Figure 3: Hyper-volume of $X(k)$ for the motor speed model using a unit ℓ_∞-ball as bound for the disturbance and a unit ℓ_2-ball for the noise.

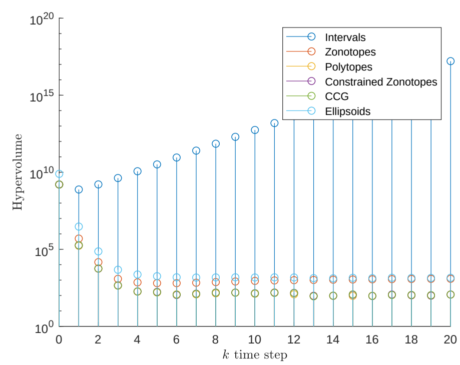

Figure 4: Hyper-volume of $X(k)$ for the suspension model using a unit ℓ_∞-ball as bound for the disturbance and a unit ℓ_∞-ball for the noise.

ellipsoids with 1.08. The best performing methods are polytopes, constrained zonotopes and CCGs with 0.105 while zonotopes achieved 0.131 at the same stage. The fact that the update step introduces conservatism when using sets with inherent symmetries means that the produced sets are larger, even though, for this model, error for vector estimates are still very similar.

In the third simulation using the *suspension* model, the *wraparound* effect due to the propagation of conservatism affects the interval computation, leading to divergence of its volume as seen in Figure 4. For this example with $n = 4$, zonotopes and ellipsoids still had similar accuracy and worse than the remaining set-membership approaches. However, we remark that the vector estimate for all methods still presents very similar error. The main drawback of using intervals for this model seems to be numerical problems as the center of the set is still a reasonable estimate for the state.

In Figure 5, it is depicted the evolution of the volume of the sets for a case that should benefit ellipsoids and CCGs as the bounds for the disturbances, noise and initial conditions are all ℓ_2-norm bounds that can be directly represented. The application of intervals is still diverging both in terms of the volume of $X(k)$ but also the vector estimate. Taking the average over the last 10 iterations to account for the part of the simulation where the volumes have converged, the worst accuracy comes from the use of zonotopes with 8.59×10^3, followed by ellipsoids with a volume of 4.15×10^3. Polytopes and constrained zonotopes are equivalent (their difference in volume is caused by the ray shooting algorithm) with 1.28×10^3 and 1.34×10^3, respectively. The best results were achieved using CCGs with 667.71 in volume, almost half of

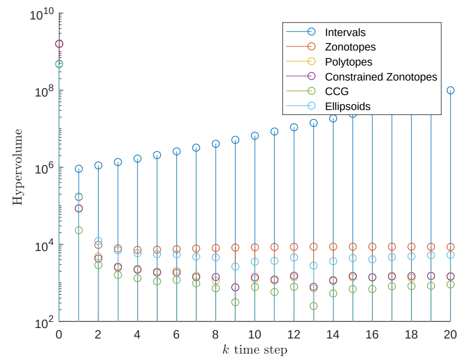

Figure 5: Hyper-volume of $X(k)$ for the linearized version of the inverted pendulum model using a unit ℓ_2-ball as bound for the disturbance and a unit ℓ_2-ball for the noise.

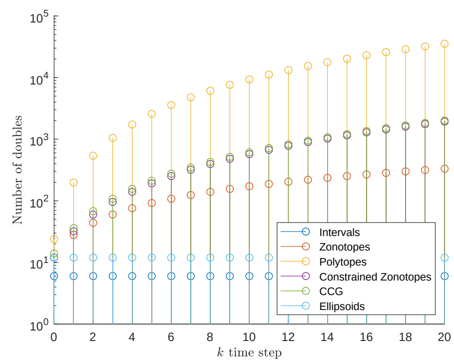

Figure 6: Number of doubles required to store the description of $X(k)$ for the aircraft pitch model using a unit ℓ_∞-ball as bound for the disturbance and a unit ℓ_2-ball for the noise.

what was achieved by constrained zonotopes.

An important question related to each set-membership approach is how it scales with the number of iterations without any reduction methods. Figure 6 reports how many double values are required to store the data structure associated with each set description for the *aircraft pitch* model. We remark that we have tested whether storing the variables as sparse structures in Matlab and then comparing the associated space overhead was helpful and only in the matrix accounting for linear constraints in polytopes we found a reduction. However, this is not significant with respect to the scalability as it still underperforms in comparison with constrained zonotopes and CCGs. In this metric, after 20 iterations, intervals are the best with 6 doubles ($2n$)

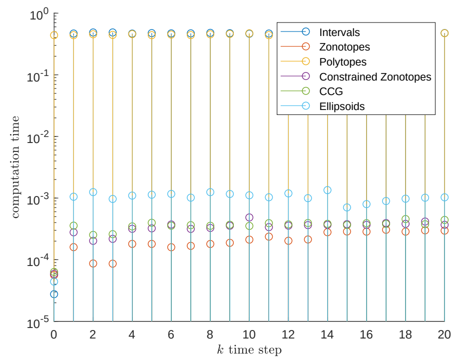

Figure 7: Computation time for the aircraft pitch model using a unit ℓ_∞-ball as bound for the disturbance and a unit ℓ_∞-ball for the noise.

followed by ellipsoids with 12 $(n^2 + n)$. Zonotopes required 332, while constrained zonotopes and CCGs used 1932 and 2012 doubles, respectively. The scalability depends on the description of the disturbances and noise bounds. The worst structure is polytopes totaling 35424 doubles.

From the previous discussion, one might expect that the performance of the methods in terms of computational time followed the same ordering in magnitude than the amount of auxiliary variables added to store the set description. This is often a point made in [5] to dismiss polytopes. Figure 7 illustrates the computation time for all methods in a favorable case where all bounds are described using ℓ_∞-norm bounds (and for example can be directly mapped in constrained zonotopes, zonotopes and CCGs). Due to the update step, intervals have a close performance to polytopes taking an average computing time 0.4495 s and 0.454 s, respectively. These methods would not be implementable in real-time given our sampling time of 0.1 s. Ellipsoids are quite efficient with an average time of 10^{-3} s. The remaining methods have very similar computational times and are ordered as CCGs $(4 \times 10^{-4}$ s), constrained zonotopes $(3 \times 10^{-4}$ s) and zonotopes (2×10^{-4}). This points towards the trend that, although having much less auxiliary variables, zonotopes offer similar performance to constrained zonotopes and CCGs but with very different characteristics in terms of accuracy given the aforementioned discussion. We would like to remark that no pre-processing of optimization programs in Yalmip was implemented and that directly using the numerical solvers can improve the performance of polytopes and intervals. There are no such considerations for the remaining structures as they do not

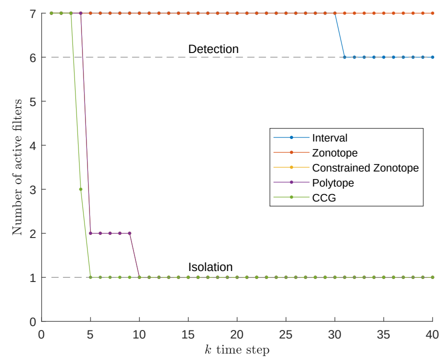

Figure 8: Evolution of the number of active filters in the bank for each of the set-membership methods.

resort to Yalmip.

4.2. Comparison for Fault Detection and Isolation

The FDI task has also been proposed for consensus systems such as in [39, 34]. In such a scenario, n agents are updating their scalar state according to a doubly stochastic (assuming that the topology is bidirectional), i.e., A_k in (3) depends on a set of parameters that account for the adjacency matrix. Following the concept in the aforementioned literature, there are no disturbances affecting the nodes and the input vector accounts for the possible faults. For that reason, we do not present results for ellipsoids since these would be degenerate ellipsoids. In order to correct that, we would have to add artificial noise or have numerical issues in the computations, which would compromise a fair comparison. In the simulation setup, $\forall k$, $\|u(k)\|_\infty \leq 1$ and matrix $B_k = \mathsf{e}_5$, where e_5 is the fifth canonical vector of size n (node 5 is faulty). We assume the topology varies randomly with a 0.4 probability of each link being established. The detector has access to the topology in each time instant (making it an LPV) and also takes noise-corrupted measures of the state values for nodes 1, 2, 7 and 8. The noise signal is assumed to satisfy $\|w(k)\|_2 \leq 0.15$. Under the assumption that only one node can be faulty, it requires to run a bank of 7 filters with 6 accounting for each possible corrupted node and M_0 for detection.

Figure 8 depicts the number of active filters in the bank for each method. As expected given the mixture of bounds in the model, the CCG method outperforms the remaining with a detection after 4 iterations and an isolation after 5. In comparison, both constrained zonotopes and polytopes were

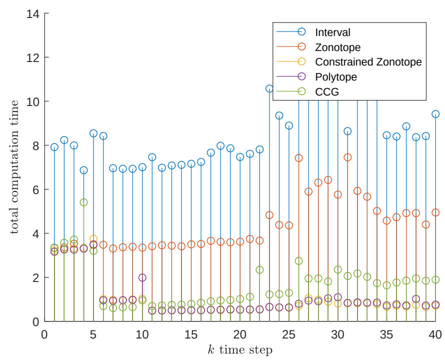

Figure 9: Sum of elapsed time for all the active filters in the bank for each of the set-membership methods.

capable of detecting a fault at iteration 5 but only isolated which node was causing it at iteration 10. Intervals also performed a detection at iteration 31 even though none of the remaining filters became empty to help the isolation. This simulation reinforces the trend observed for the volume evolution in the state estimation problem.

We also report in Figure 9 the total computation time for the entire bank of filter of each method. Since zonotopes and intervals were not successful in the FDI task, the main conclusion pertains to how CCGs performed against constrained zonotopes and polytopes. Interestingly, since both zonotopes and polytopes have linear description, the feasibility problems were solved in roughly the same time over all time instants with an average of 1.0435 s and 1.0832 s, respectively. The CCGs took additional time due to the quadratic constraints with an average 1.7034 seconds. Given the performance of the filter banks, it is not expected that these mechanisms work in real-time and the added CCGs accuracy is beneficial to the task.

4.3. Comparison for Collision Avoidance for Autonomous Vehicles

In this section, we recover the example considering unicycle dynamics described in [14]. The vehicle schematic representation is given in Figure 10 and has the following dynamics in discrete-time:

$$\begin{bmatrix} p_i \\ q_i \end{bmatrix}(k+1) = \begin{bmatrix} p_i \\ q_i \end{bmatrix}(k) + \mathrm{Ts}\ A_i(\theta_i) \begin{bmatrix} v_i \\ w_i \end{bmatrix}(k)$$

where the state (p_i, q_i) identify the position of the front of the ith vehicle and the inputs (v_i, w_i) account for the linear velocity and rotation. Moreover,

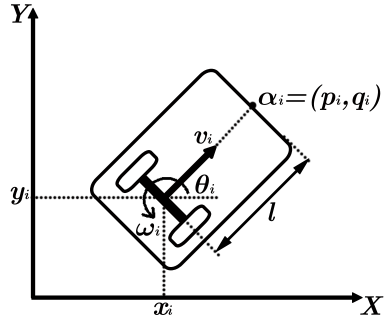

Figure 10: Schematic of the unicycle model for the vehicles.

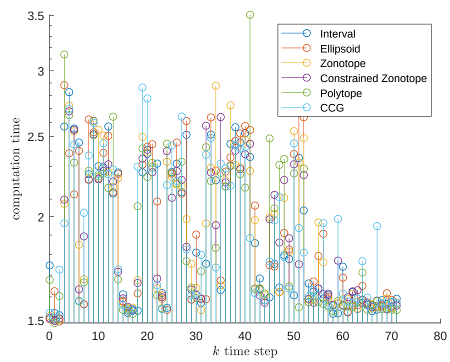

Figure 11: Computation time to detect potential collisions with a polytopic and ellipsoidal obstacles.

$\mathrm{Ts} = 0.1$ stands for the sampling time, θ_i (we omit the time dependence in k for a more compact presentation) for the orientation and matrix $A_i(\theta_i)$ is given as:

$$A_i(\theta_i) = \begin{bmatrix} \cos\theta_i & -l\sin\theta_i \\ \sin\theta_i & l\cos\theta_i \end{bmatrix}.$$

In this simulation, we consider a single vehicle and, assuming that its orientation $\theta_1(k)$ can be measured at each time instant k, matrix A_1 can be computed and the model falls under the umbrella an LPV as in (3).

The control law applied to the vehicle to track a trajectory is:

$$\begin{bmatrix} v_i(k) \\ w_i(k) \end{bmatrix} = \frac{A_i^{-1}(\theta_i)}{\mathrm{Ts}} \left(\tau(k+1) - \frac{\tau(k)}{2} - 0.5 \begin{bmatrix} p_i(k) \\ q_i(k) \end{bmatrix} + d(k) \right)$$

where $\tau(k)$ accounts for the discrete sequence of waypoints in the trajectory.

In Figure 11, it is presented the time taken by each of the methods, encompassing constructing the set-valued estimate for the position, construct the feasibility problem using Yalmip and finding whether there is an intersection with any of the obstacles using Mosek. As depicted in Figure 11, the values in each iteration are clustered which indicate that the various set-membership techniques are rather equivalent in terms of performance for this task. In terms of average computation time, we have the following results: constrained zonotope (1.9181 s), intervals (1.9243 s), polytopes (1.9428 s), CCGs (1.9459 s), zonotopes (1.9482 s) and ellipsoids (1.963 s).

5. Conclusions and Future Work

In this paper, a comparison was presented related to the state-of-the-art in set-membership techniques both for state estimation, fault detection and isolation and collision avoidance. Whenever faced with a Linear Parameter-Varying system, given the absence of bilinear constraints in the set definition, the produced sets will remain convex assuming bounds for disturbances, noise and initial conditions are also convex. The current proposals allow to define various sets with different properties and in polytopes, constrained zonotopes and CCGs, all required set operations can be performed in closed form, which substantially increases their performance.

In the state estimation realm, we have covered various physical systems in order to assess how the proposals fair in terms of accuracy of an estimate, hyper-volume of the entire set, number of doubles required in the data structure and computation time. The following conclusions can be drawn from this study:

- intervals, zonotopes and ellipsoids are rather inefficient due to the lack of a closed-form expression for the update step;

- intervals, zonotopes and ellipsoids present poor accuracy given the required symmetries of the sets (which in turn saves storing space);

- constrained zonotopes are a better alternative to polytopes by presenting a similar accuracy but storing a much smaller number of auxiliary variables;

- CCGs have a negligible increment in the required computation time in comparison to constrained zonotopes but offer better accuracy especially when dealing with ℓ_2 bounds or a combination of polytopic and ℓ_2 bounds.

In the fault detection and isolation experiment, the aforementioned conclusions can be better visualized since isolation was done in half the number of time instants than when using constrained zonotopes. We also remark that CCGs have the ability to represent unbounded sets, which are particularly interesting when dealing with bearings measurements in autonomous vehicles. The main avenue of future research would be to perform a similar study for techniques that can be applied to linear systems in the presence of uncertainties. Results in this regard could clarify the literature related to a subset of nonlinear systems known as uncertain Linear Parameter-Varying models.

Acknowledgement

This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through project FirePuma (https://doi.org/10.54499/PCIF/MPG/0156/2019), through LARSyS FCT funding (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIDP/50009/2020, and 10.54499/UIDB/50009/2020) and through COPELABS, University Lusófona project UIDB/04111/2020.

References

[1] Fahed Abdallah, Amadou Gning, and Philippe Bonnifait. Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, 44(3):807 – 815, 2008.

[2] T. Alamo, J.M. Bravo, and E.F. Camacho. Guaranteed state estimation by zonotopes. *Automatica*, 41(6):1035 – 1043, 2005.

[3] Amr Alanwar, Victor Gaßmann, Xingkang He, Hazem Said, Henrik Sandberg, Karl H. Johansson, and Matthias Althoff. Privacy-preserving set-based estimation using partially homomorphic encryption. *European Journal of Control*, 71:100786, 2023.

[4] Matthias Althoff and John M Dolan. Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, 30(4):903–918, 2014.

[5] Matthias Althoff and Jagat Jyoti Rath. Comparison of guaranteed state estimators for linear time-invariant systems. *Automatica*, 130:109662, 2021.

[6] Franco Blanchini and Stefano Miani. *Set-theoretic methods in control*, volume 78. Springer, 2008.

[7] F.L. Chernousko. Ellipsoidal state estimation for dynamical systems. *Nonlinear Analysis: Theory, Methods & Applications*, 63(5):872 – 879, 2005. Invited Talks from the Fourth World Congress of Nonlinear Analysts (WCNA 2004).

[8] Luigi Chisci, Andrea Garulli, and Giovanni Zappa. Recursive state bounding by parallelotopes. *Automatica*, 32(7):1049–1055, 1996.

[9] C. Combastel. A state bounding observer based on zonotopes. In *2003 European Control Conference (ECC)*, pages 2589–2594, 2003.

[10] Christophe Combastel. Zonotopes and kalman observers: Gain optimality under distinct uncertainty paradigms and robust convergence. *Automatica*, 55:265–273, 2015.

[11] Christophe Combastel. An Extended Zonotopic and Gaussian Kalman Filter (EZGKF) merging set-membership and stochastic paradigms: Toward non-linear filtering and fault detection. *Annual Reviews in Control*, 42:232–243, 2016.

[12] Michele Conforti, Marco Di Summa, and Yuri Faenza. Balas formulation for the union of polytopes is optimal. *Mathematical Programming*, 180(1):311–326, Mar 2020.

[13] Abhishek Halder. On the parameterized computation of minimum volume outer ellipsoid of minkowski sum of ellipsoids. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4040–4045, 2018.

[14] D. E. Hernández-Mendoza, G. R. Penaloza-Mendoza, and E. Aranda-Bricaire. Discrete-time formation and marching control of multi-agent robots systems. In *2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control*, pages 1–6, Oct 2011.

[15] C. Hoffmann and H. Werner. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on Control Systems Technology*, 23(2):416–433, 2015.

[16] A. Agung Julius and George J. Pappas. Trajectory based verification using local finite-time invariance. In Rupak Majumdar and Paulo Tabuada, editors, *Hybrid Systems: Computation and Control*, pages 223–236, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[17] Shreyas Kousik, Adam Dai, and Grace X. Gao. Ellipsotopes: Uniting ellipsoids and zonotopes for reachability analysis and fault detection. *IEEE Transactions on Automatic Control*, pages 1–13, 2022.

[18] A.B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis: internal approximation. *Systems & Control Letters*, 41(3):201–211, October 2000.

[19] Alex A. Kurzhanskiy and Pravin Varaiya. Ellipsoidal toolbox (et). In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1498–1503, 2006.

[20] W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61(1):47–67, March 1998.

[21] Tuochen Li, Cunxi Zheng, Youdao Ma, and Zhenhua Wang. Interval estimation for discrete-time linear time-delay systems based on state augmentation. *European Journal of Control*, 72:100823, 2023.

[22] Muhammad Umar B. Niazi, Amr Alanwar, Michelle S. Chong, and Karl H. Johansson. Resilient set-based state estimation for linear time-invariant systems using zonotopes. *European Journal of Control*, 74:100837, 2023. 2023 European Control Conference Special Issue.

[23] Masoud Pourasghar, Vicenç Puig, and Carlos Ocampo-Martinez. Comparison of set-membership and interval observer approaches for state estimation of uncertain systems. In *2016 European Control Conference (ECC)*, pages 1111–1116. IEEE, 2016.

[24] Vignesh Raghuraman and Justin P. Koeln. Set operations and order reductions for constrained zonotopes. *Automatica*, 139:110204, 2022.

[25] B. S. Rego, D. M. Raimondo, and G. V. Raffo. Set-based state estimation of nonlinear systems using constrained zonotopes and interval arithmetic. In *2018 European Control Conference (ECC)*, pages 1584–1589, 2018.

[26] Francisco Rego, Elwin de Weerdt, Eddy van Oort, Erik-Jan van Kampen, Qiping Chu, and António M Pascoal. Determination of inner and outer bounds of reachable sets through subpavings. *Mathematics in Computer Science*, 8(3):425–442, 2014.

[27] Rafael Ribeiro, Daniel Silvestre, and Carlos Silvestre. A rendezvous algorithm for multi-agent systems in disconnected network topologies. In *2020 28th Mediterranean Conference on Control and Automation (MED)*, pages 592–597, 2020.

[28] Rafael Ribeiro, Daniel Silvestre, and Carlos Silvestre. Decentralized control for multi-agent missions based on flocking rules. In José Alexandre Gonçalves, Manuel Braz-César, and João Paulo Coelho, editors, *CONTROLO 2020*, pages 445–454, Cham, 2021. Springer International Publishing.

[29] Sadra Sadraddini and Russ Tedrake. Linear encodings for polytope containment problems. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4367–4372, 2019.

[30] F. Schweppe. Recursive state estimation: Unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, 13(1):22–28, 1968.

[31] Joseph K. Scott, Davide M. Raimondo, Giuseppe Roberto Marseglia, and Richard D. Braatz. Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, 69:126 – 136, 2016.

[32] Jeff S. Shamma and Michael Athans. Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27(3):559 – 564, 1991.

[33] J.S. Shamma and Kuang-Yang Tu. Set-valued observers and optimal disturbance rejection. *IEEE Transactions on Automatic Control*, 44(2):253 –264, feb 1999.

[34] 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*, pages 4373–4378, Dec 2013.

[35] Daniel Silvestre. Accurate guaranteed state estimation for uncertain lpvs using constrained convex generators. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4957–4962, 2022.

[36] Daniel Silvestre. Constrained convex generators: A tool suitable for set-based estimation with range and bearing measurements. *IEEE Control Systems Letters*, 6:1610–1615, 2022.

[37] Daniel Silvestre. Set-valued estimators for uncertain linear parameter-varying systems. *Systems & Control Letters*, 166:105311, 2022.

[38] Daniel Silvestre. Exact set-valued estimation using constrained convex generators for uncertain linear systems. *IFAC-PapersOnLine*, 56(2):9461–9466, 2023. 22nd IFAC World Congress.

[39] Daniel Silvestre, Paulo Rosa, João P. Hespanha, and Carlos Silvestre. Finite-time average consensus in a byzantine environment using set-valued observers. In *2014 American Control Conference*, pages 3023–3028, 2014.

[40] Daniel Silvestre, Paulo Rosa, João P. Hespanha, and Carlos Silvestre. Fault detection for lpv systems using set-valued observers: A coprime factorization approach. *Systems & Control Letters*, 106:32 – 39, 2017.

[41] Daniel Silvestre, Paulo Rosa, João P. Hespanha, and Carlos Silvestre. Set-based fault detection and isolation for detectable linear parameter-varying systems. *International Journal of Robust and Nonlinear Control*, 27(18):4381–4397, 2017.

[42] Daniel Silvestre, Paulo Rosa, João P. Hespanha, and Carlos Silvestre. Stochastic and deterministic fault detection for randomized gossip algorithms. *Automatica*, 78:46 – 60, 2017.

[43] Daniel Silvestre, Paulo Rosa, João P. Hespanha, and Carlos Silvestre. Self-triggered and event-triggered set-valued observers. *Information Sciences*, 426:61 – 86, 2018.

[44] Daniel Silvestre, Paulo Rosa, and Carlos Silvestre. Distinguishability of discrete-time linear systems. *International Journal of Robust and Nonlinear Control*, 31(5):1452–1478, 2021.

[45] W. Tang, Z. Wang, Y. Wang, T. Raïssi, and Y. Shen. Interval estimation methods for discrete-time linear time-invariant systems. *IEEE Transactions on Automatic Control*, 64(11):4717–4724, 2019.

[46] Wentao Tang, Zhenhua Wang, Ye Wang, Tarek Raïssi, and Yi Shen. Interval estimation methods for discrete-time linear time-invariant systems. *IEEE Transactions on Automatic Control*, 64(11):4717–4724, 2019.

[47] Rihab El Houda Thabet, Tarek Raïssi, Christophe Combastel, Denis Efimov, and Ali Zolghadri. An effective method to interval observer design for time-varying systems. *Automatica*, 50(10):2677 – 2684, 2014.

[48] J. Wan, S. Sharma, and R. Sutton. Guaranteed state estimation for nonlinear discrete-time systems via indirectly implemented polytopic set computation. *IEEE Transactions on Automatic Control*, 63(12):4317–4322, 2018.

[49] Shaocheng Wang and Wei Ren. On the Convergence Conditions of Distributed Dynamic State Estimation Using Sensor Networks: A Unified Framework. *IEEE Transactions on Control Systems Technology*, 26(4):1300–1316, July 2018.

[50] Xuejiao Yang and Joseph K Scott. A comparison of zonotope order reduction techniques. *Automatica*, 95:378–384, 2018.

[51] Gang Zheng, Denis Efimov, Francisco Javier Bejarano, Wilfrid Perruquetti, and Haoping Wang. Interval observer for a class of uncertain nonlinear singular systems. *Automatica*, 71:159–168, September 2016.