European Journal of Control 81 (2025) 101163

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.sciencedirect.com/journal/european-journal-of-control

Cooperative navigation using constrained convex generators

Francisco Rego [a,b,c], Daniel Silvestre [d,b,c,*]

[a] EET-IPLuso, CI-Luso, Instituto Politécnico da Lusofonia, Lisbon, Portugal
[b] Centre of Technology and Systems/UNINOVA, School of Science and Technology—NOVA University of Lisbon, Caparica, Portugal
[c] Institute for Systems and Robotics, Lisbon, Portugal
[d] School of Science and Technology from the NOVA University of Lisbon (FCT/UNL), Caparica, Portugal

ARTICLE INFO

Recommended by T. Parisini

Keywords:
Cooperative navigation
Constrained convex generators
Set-valued observers
Range and bearing-based navigation
Distributed estimation

ABSTRACT

In this article, we suggest a technique of set-based cooperative navigation for a fleet of vehicles under communication constraints. In the suggested approach, only the distances and bearings to other vehicles or known locations can be measured. Each vehicle in the fleet must determine its location and the positions of the other vehicles based on the measurements as well as information shared among the vehicles. Since the measurement set may be nonconvex, it must be approximated by a convex set. To address this issue, we employ constrained convex generators, which is a generalization of the definition for constrained zonotopes that allows ℓ_2 unit norm balls and convex cones. To keep the amount of exchanged information low the vehicles transmit an ellipse containing the state estimate to the other vehicles. The obtained estimates are worst-case bounds for the true position, which is important in certain applications such as collision avoidance. The computed sets are then applied to vehicle control algorithms, taking into account the positions of all the agents. Through numerical simulations, we illustrate the application of this method to the problem of cooperative navigation of autonomous underwater vehicles and the problem of fire detection with multiple UAVs.

1. Introduction

The creation of sophisticated cooperative marine robotic systems that can function entirely autonomously without being closely monitored by human operators is of great interest. Applications range from scientific surveys (such as seabed mapping) in remote locations to inspections of industrial underwater infrastructures and the collection of underwater images at sites of public interest (such as archaeological surveys). To achieve these goals efficiently, autonomous underwater vehicles (AUVs) are an extremely important tool. When conducting missions as a fleet, cooperative navigation is a crucial method to be employed that also aids in providing vehicles with position estimations as well as for geo-referencing the collected data.

Due to the strong attenuation of electromagnetic waves in the underwater environment, medium- and long-range communications are typically carried out using acoustic signals, whose range can vary from hundreds of metres to hundreds of kilometres depending on the signal frequency (lower frequency yields greater range) (González-García et al., 2020). Additionally, the modulation technique and frequency (higher frequencies result in greater bit rates) affect the bit rate (Benson et al., 2010). Bit rates for commercial modems range from 40 to 15 000 bps.

It is sometimes necessary for a fleet of several vehicles surveying the ocean floor to operate in a certain formation with respect to one another. To achieve this goal, each vehicle must have access to the relative locations of its neighbours in a particular formation to achieve this objective (Ren & Beard, 2008). In the underwater scenario, data from one vehicle may be broadcast to all vehicles over the supporting acoustic communication network in applications where a group of vehicles operate in proximity. As a result, we obtain a full graph.

In this paper, we propose a method of cooperative navigation for a fleet of AUVs for formation control under bandwidth restrictions imposed by the acoustic communications medium. We picture a situation in which one vehicle is equipped with a GNSS positioning device and each of the other vehicles is outfitted with ultra-short baseline (USBL) devices that measure distance and bearing to the other vehicles in the formation. This study proposes a cooperative navigation method of computing, for each vehicle, a set containing its position and of its neighbours using range and bearing measurements, as well as information shared among the vehicles.

Since distributed state estimation has gained popularity in recent years, there is a significant body of literature on the subject. Many versions of the distributed Kalman filter have been suggested (Battistelli & Chisci, 2016; Olfati-Saber, 2007). In Basu and Yoon (2019), Mitra and

* Corresponding author.Largo da Torre, 2829-516 Caparica, Portugal.
E-mail address: dsilvestre@fct.unl.pt (D. Silvestre).

https://doi.org/10.1016/j.ejcon.2024.101163
Received 26 April 2024; Received in revised form 20 August 2024; Accepted 23 November 2024
Available online 2 December 2024

Sundaram (2018) and Rego (2023) distributed Luenberger observers for linear systems are proposed, where similarly to distributed Kalman filters, they rely on consensus dynamics (Silvestre, Hespanha, & Silvestre, 2019) using the estimates. In Rego, Pascoal, Aguiar, and Jones (2019), it can be found a comprehensive study of distributed estimate methods.

Algorithms that calculate sets that at any given time contain all potential state values are called set-based observers of a dynamical system. The surveys in Althoff and Rath (2021) and Pourasghar, Puig, and Ocampo-Martinez (2019) show that there has been a lot of interest in the topic of designing set-based observers. One important choice when designing set-valued observers is how to represent the sets. One possibility is the use of intervals (Bako, Ndiaye, & Blanco, 2022) or ellipsoids as in Chernousko (2005) and Kurzhanski and Varaiya (2000). Another possible choice is to use zonotopes, which have a reduced wrapping effect in comparison with the intervals and ellipsoids (Alamo, Bravo, & Camacho, 2005; Combastel, 2015; Wang & Ren, 2018). Polytopes can both be represented in semi-explicit format (Silvestre, Rosa, Hespanha, & Silvestre, 2017), Constrained Zonotopes (i.e., as the linear map of a generator set) (Scott, Raimondo, Marseglia, & Braatz, 2016) or based on point operations using their vertices (Silvestre, 2022b). Constrained convex generators are a more recent solution that unifies all these set representations (Silvestre, 2022a).

The extension of set-based observers to the distributed setting has received great attention lately (Alanwar, Rath, Said, Johansson, & Althoff, 2020). The work in García, Rubio, Orihuela, Millán, and Ortega (2017) and Kieffer (2009) contains distributed estimation algorithms that use zonotopes. In Orihuela, Millán, Roshany-Yamchi, and García (2018) and Orihuela, Roshany-Yamchi, García, and Millán (2017), the authors extended the approach in Combastel (2015) to a distributed framework with network-based challenges and provided a Kalman-based set-membership observer. The work in Combastel and Zolghadri (2018) contains an extension to the distributed setting of Combastel (2016), which merges the set membership and the stochastic paradigms. The work in Wang, Alamo, Puig, and Cembrano (2018) suggests a distributed zonotope-based estimator that applies to a particular family of systems with state interconnections. A distributed set-valued observer was suggested in Silvestre, Rosa, Hespanha, and Silvestre (2015) which makes use of a coprime factorization to handle relative measurements. A similar problem was addressed in Silvestre, Rosa, Hespanha, and Silvestre (2014) which uses set-valued observers to address the problem of average consensus in a Byzantine environment. Recently (García, Orihuela, Millán, Rubio, & Ortega, 2020) studied the application of distributed set-based observers to vehicle formation control.

In this paper, we aim to refine set-based cooperative navigation methods with communication limitations, which are critical for applications like collision avoidance and fault detection. Therefore, we propose a method for cooperative set-based navigation with limited communications. Only the distances and bearings to other vehicles may be measured using the proposed method. Based on the measurements and information communicated among the vehicles, each vehicle in the fleet must determine its location as well as the positions of the other vehicles. Since the measurement set must be approximated by a convex set, constrained convex generators (CCGs), which are an extension of the definition for constrained zonotopes, are used to handle this problem.

The paper introduces several key innovations.

- We propose a set-based cooperative navigation algorithm tailored for underwater navigation and drone fire detection scenarios. Unlike traditional methods such as Dong, Yu, Shi, and Zhong (2014) and Li, Xie, and Yan (2016), our approach, similar to García et al. (2020), is set-based. This means we acquire information on all possible states of the system.
- We employ constrained convex generators as set descriptors, providing a more precise representation of the sets derived from measurements. Unlike (García et al., 2020), this method allows for greater accuracy in describing the sets compared to the traditional use of constrained zonotopes.

1.1. Notation

Let I_n be the identity matrix of size n, and let $\mathbf{0}_n$ stand for the n-dimensional array of zeros and $\mathbf{1}_n$ denote the n-dimensional array of ones. The vector e^i consists of zeros except in the ith position which contains a 1. Whenever the index is omitted $\mathbf{0}$ denotes a matrix of zeros whose size can be inferred from the context. The transpose of a vector v is written by $v^\intercal$, and the Euclidean norm for a vector x is denoted by $\|x\|_2 := \sqrt{x^\intercal x}$. Additionally, $\|x\|_\infty := \max_i |x(i)|$, where $x(i)$ is the ith element of x. The intersection after applying a matrix R to the first set is represented by $\cap_R$, the Minkowski sum of two sets by $\oplus$, the Cartesian product by $\times$, the Kronecker product by $\oplus$, and the modulus operator or remainder after division is denoted as $\mathrm{mod}(\cdot)$.

2. Method

2.1. Mathematical background

2.1.1. Constrained convex generators (CCGs)

We first introduce the definition and the main operations of CCGs. Definition 1 provide a formal description of CCGs.

Definition 1 (*Constrained Convex Generators*). $\mathcal{Z} \subset \mathbb{R}^n$ is defined by the tuple $(G, c, A, b, \mathfrak{C})$ with $G \in \mathbb{R}^{n_c \times n_g}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{n_c \times n_g}$, $b \in \mathbb{R}^{n_c}$, and $\mathfrak{C} := \left\{ C_1, C_2 \ldots, C_{n_p} \right\}$ such that:

$$\mathcal{Z} = \{G\xi + c : A\xi = b, \xi \in C_1 \times \cdots \times C_{n_p}\}. \tag{1}$$

CCGs are a very general form of representing sets since ℓ_p norm balls, norm cones, among others can be represented directly. This entails that no approximation is required to represent ellipsoidal shapes, polytopes or even unbounded sets, which would otherwise introduce conservatism in the case of constrained zonotopes or polytopes. For instance, polytopes can be represented as

$$\mathcal{X} = \left(G, c, A, b, \|\xi\|_\infty \le 1\right) \tag{2}$$

and ellipsoids are defined as

$$\mathcal{X} = \left(G, c, [\,], [\,], \|\xi\|_2 \le 1\right) \tag{3}$$

Other types of sets can also be described as CCGs such as ellipsotopes, intervals, or zonotopes. For more information on CCGs, the reader is referred to Silvestre (2022a). The usual operations such as linear maps, Minkowski sum, and intersection are well-defined for CCGs and can be computed as in Definition 2.

Definition 2. Consider three Constrained Convex Generators (CCGs) as in Definition 1:

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

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:

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

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

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

2.1.2. Guaranteed state estimation

The problem of guaranteed state estimation in discrete-time LTI systems can be formulated as the problem of finding a set of possible state values given measurements, disturbance, noise, and initial state bounds. The model is provided by:

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{4a}$$

$$y_k = Cx_k + v_k, \tag{4b}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^{n_u}$, $w_k \in \mathbb{R}^n$, $y_k \in \mathbb{R}^{n_y}$, and $v_k \in \mathbb{R}^{n_y}$ represent the system state, input, disturbance signal, output, and noise, respectively. The problem this article addresses can be summed up as follows:

Problem 1. How to calculate a set X_k that ensures that $x_k \in X_k, \forall k \geq 0$, given compact convex sets X_0, V, and W, such that $x_0 \in X_0$, $v_k \in V$ and $w_k \in W$ for all $k \geq 0$, and measurements y_k.

Given Definition 2, one may solve Problem 1 recursively, since given a set $X_k \subset \mathbb{R}^n$ such that $x_k \in X_k$ and a measurement y_k, the set $X_{k+1} \subset \mathbb{R}^n$ such that $x_{k+1} \in X_{k+1}$ can be computed as

$$X_{k+1} = \left(AX_k \oplus W + Bu_k\right) \cap_C (y_k - V). \tag{5}$$

In this implementation, we assume that the sets W and V are CCGs, resulting in less conservatism than other representations. Specifically in this paper, we consider a constant description for the disturbance and noise sets:

$$W := \left(G_w, c_w, [\,], [\,], \mathfrak{C}_w\right), \tag{6a}$$

$$V := \left(G_v, c_v, [\,], [\,], \mathfrak{C}_v\right). \tag{6b}$$

2.1.3. Range and bearing measurements CCG

Central to this work is the representation of the set of admissible positions when receiving range and bearing measurements with a certain error, which as shown in Silvestre (2022a), is done accurately with CCGs. Given a certain 2D position $p \in \mathbb{R}^2$ and a measurement of that position

$$y^{br} := \begin{bmatrix} y^r \\ y^b \end{bmatrix}, \tag{7}$$

where

$$y^r := \|p\| + v^r, \tag{8}$$

$$y^b := \mathrm{ang}(p) + v^b, \tag{9}$$

the operator $\mathrm{ang}(p)$ returns the angle of vector p in polar coordinates, and the measurement noise variables v^r and v^b satisfy $r_l \leq v^r \leq r_u$ and $b_l \leq v^b \leq b_u$, we would like to compute a CCG $Y(y^{br})$ such that $p \in Y(y^{br})$. The set of possible positions can be represented as the blue area in Fig. 1.

Following Silvestre (2022a), first, we calculate the four points that are produced by the minimum and maximum angles at the intersection of each circle, resulting in two outer points in the outer circle (points 1 and 3 in Fig. 1) and two inner points (points 2 and 4 in Fig. 1). The coordinates of each point can be expressed as $\rho \left[\cos\alpha \;\; \sin\alpha\right]$, where ρ is selected from the set $\{y^r - r_l, y^r + r_u\}$ and α is chosen from the set $\{y^b - b_l, y^b + b_u\}$. Additionally, we require a fifth point with $\alpha = (2y^b - b_l + b_u)/2$ and the largest possible range $\rho = y^r + r_u$. The remaining line equations and the trapezoidal shape may then be written as $Mx \leq m$ by finding the line equation that is parallel to the outer points and passes through the fifth point. We derive the Constrained Zonotope (G, c, A, b) by using the formula from Theorem 1 in Scott et al. (2016), which is identical to the CCG representation $\mathcal{Z}_{trap} = (G, c, A, b, \{\mathcal{B}_\infty\})$ where $\mathcal{B}_\infty$ is the unit ℓ_∞-ball. An illustration of $\mathcal{Z}_{trap}$ is given as the red area in Fig. 2.

The CCG $\left((y^r + r_u)I_2, \mathbf{0}, \mathbf{0}, 0, \{\mathcal{B}_2\}\right)$, where $\mathcal{B}_2$ is the unit 2-ball, gives the outside circle. The measurement set $Y(y^{br})$ is therefore provided by:

$$Y(y^{br}) = \mathcal{Z}_{trap} \cap_{I_2} \left((y^r + r_u)I_2, \mathbf{0}, \mathbf{0}, 0, \{\mathcal{B}_2\}\right), \tag{10}$$

and is represented as the blue area of Fig. 3.

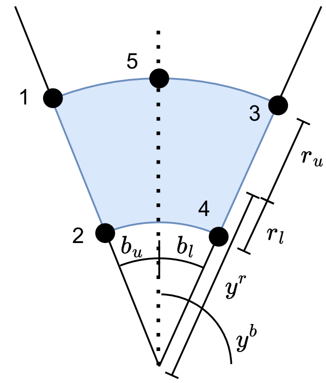

Fig. 1. Set of possible positions given a measurement y^{br}.

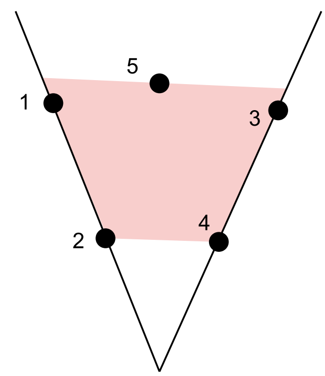

Fig. 2. CZ enclosure of the measurement set.

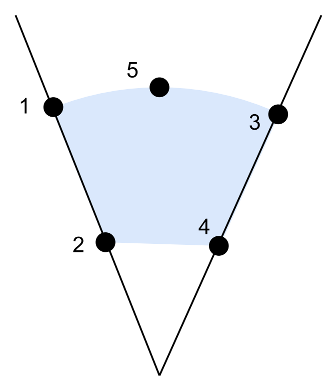

Fig. 3. CCG enclosure of the measurement set.

2.2. Problem statement

In this paper, we aim to adapt the CCG representation in Silvestre (2022a) to the problem of cooperative navigation in the realistic scenarios of a group of AUVs moving in formation following a surface vehicle and a group of drones moving in formation to detect a fire source.

2.2.1. Cooperative underwater navigation problem

We assume that each element of the group of AUVs has ultra-short baseline (USBL) devices that measure the ranges and bearings of the other vehicles. In addition, one of the AUVs, which we will refer to as vehicle 1, is at the surface and has a GPS that measures its location. The problem we attempt to solve in this study is estimating at each vehicle the set of possible locations of all vehicles in order to avoid having to consider their control law as uncertain in the formation control algorithm when using these estimates. The formation control issue we address, in particular, corresponds to a scenario in which vehicle 1 must follow a pre-defined trajectory and the other vehicles must maintain pre-defined relative positions to the vehicle 1 and one another. The vehicles are assumed to move on a 2D plane. However, this assumption may be lifted by assuming the vehicles can measure their depth. Furthermore, for the purposes of designing the observer, we assume that the vehicles can be modelled as single integrators with the velocity as a control input as follows:

$$\dot{p}^i = u^i + w^i, \tag{11}$$

where $i \in \{1,\dots,N\}$ is a vehicle index that identifies the vehicle in the fleet, N is the number of vehicles, $p^i \in \mathbb{R}^2$ is the position of the vehicle, $u^i \in \mathbb{R}^2$ is the control input which is a velocity reference that the vehicle must follow and $w^i \in \mathbb{R}^2$ is the disturbance which is the velocity tracking error. The velocity reference u^i serves as a reference to a velocity tracking system that issues thrust commands to the actuators.

We assume that for each Δ_t unit of time, one of the vehicles transmits information while the remaining vehicles receive that information and compute the range and bearing to that vehicle. Moreover, we assume that the vehicles have piecewise constant velocity reference signals which are sampled through a zero-order hold with an inter-sample interval of Δ_t units of time. Therefore, defining a time index $k \in \mathbb{Z}$ the discretized dynamics of vehicle i can be expressed as

$$x^i_{k+1} = A_l x^i_k + B_l u^i_k + w^i_k, \tag{12}$$

where

$$x^i_k := p^i_k, \quad A_l := I_2, \quad B_l := I_2\Delta_t, \tag{13}$$

where $p^i_k \in \mathbb{R}^2$ is the position of vehicle i at time $t = k\Delta_t$, $u^i_k \in \mathbb{R}^2$ is the control input of vehicle i, which serves as a velocity reference for vehicle i for $k\Delta_t \le t < (k+1)k\Delta_t$, and w^i_k is the disturbance or tracking error.

Combining the states of all the vehicles, the dynamics of the fleet can be expressed as

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{14}$$

where

$$x_k := \begin{bmatrix} x^1_k \\ \vdots \\ x^N_k \end{bmatrix}, \quad u_k := \begin{bmatrix} u^1_k \\ \vdots \\ u^N_k \end{bmatrix}, \quad w_k := \begin{bmatrix} w^1_k \\ \vdots \\ w^N_k \end{bmatrix}, \tag{15}$$

and $A := I_N \otimes A_l$ and $B := I_N \otimes B_l$.

The measurement vector of a vehicle i when receiving data from vehicle j can be expressed as

$$y^{i,j}_k = h^{i,j}(x_k) + v^{i,j}_k, \tag{16}$$

where $v^{i,j}_k$ is the measurement noise and is considered to be constrained as

$$\begin{bmatrix} r_l \\ b_l \end{bmatrix} \le v^{i,j}_k \le \begin{bmatrix} r_u \\ b_u \end{bmatrix} \tag{17}$$

where the inequality is elementwise. The measurement functions for vehicles $i \in \{1,\dots,N\}$ are given by

$$h^{i,j}(x_k) := \begin{bmatrix} \|p^i_k - p^j_k\| \\ \text{ang}\left(p^i_k - p^j_k\right) \end{bmatrix}. \tag{18}$$

Moreover, it is assumed that vehicle 1 measures its own position since it is assumed to be at the surface with a GNSS positioning device. Therefore at every sample time vehicle 1 measures

$$y^p_k = p^1_k + v^p_k, \tag{19}$$

where v^p_k is the measurement noise and is considered to be constrained as $\|v^p_k\| \le p_u$.

2.2.2. Drone fire detection problem

The problem of fire detection consists of several drones equipped with cameras and GPS communicating among themselves to determine the position of a fire source. In this case, the cameras determine the approximate distance and bearing to the fire with a large margin of error, while the drones communicate their own position and other information to improve the accuracy of the position estimate of the fire source.

Defining the position of the drone i relative to the fire source as p^i_k and given that the vehicle has access to its own GPS position, determining the set of possible locations of the fire source amounts to computing the set of admissible p^i_k for each drone given the measurements and data received.

For fire detection, the measurement vector of a vehicle i when receiving data from vehicle j can be expressed as

$$y^{i,j}_k = p^i_k - p^j_k + v^{i,j}_k, \tag{20}$$

where $v^{i,j}_k$ is the measurement noise and is considered to be constrained as $\|v^{i,j}_k\| \le p_u$.

At every sample time each vehicle measures

$$y^i_k = \begin{bmatrix} \|p^i_k\| \\ \text{ang}\left(p^i_k\right) \end{bmatrix} + v^i_k, \tag{21}$$

where v^i_k is the measurement noise and is considered to be constrained as

$$\begin{bmatrix} r_l \\ b_l \end{bmatrix} \le v^i_k \le \begin{bmatrix} r_u \\ b_u \end{bmatrix}. \tag{22}$$

It is important to note that cooperation is employed to enhance each drone's estimate of the fire source position relative to its own location. While GPS provides accurate positioning, the cameras used for detecting the fire source have limited precision. Thus, the drones benefit from cooperation, as demonstrated in the results section.

The assumption of 2D motion is common, as it presumes that an altitude controller is active, ensuring the drones maintain a constant, known altitude.

2.3. Velocity tracking

The navigation algorithms assume that velocity is directly controlled with a certain disturbance, as described in (11).

In this work, we assume that this condition holds for aerial vehicles. However, due to the slower dynamics of AUVs, we will simulate velocity tracking using a velocity controller, which is explained next. Despite this, the navigation algorithm maintains the assumption that (11) is true.

2.3.1. AUV velocity tracking

At each time sample, a velocity reference is issued that must be tracked by each AUV. We assume that each AUV moves in a horizontal plane and is equipped with an attitude and heading reference system (AHRS) which provides yaw measurements in real-time.

The kinematic equations for the simplified 3 DOF model of the vehicle in the horizontal plane are:

$$\dot{x} = u\cos\theta - v\sin\theta, \tag{23}$$

$$\dot{y} = u\sin\theta + v\cos\theta, \tag{24}$$

$$\dot{\theta} = r. \tag{25}$$

The dynamics equations are the following:

$$m_u\dot{u} - m_v vr = \tau_u - d_u u - d_{u|u|}u|u|, \tag{26a}$$

$$m_v\dot{v} + m_u ur = -d_v v - d_{v|v|}v|v|, \tag{26b}$$

$$m_r\dot{r} - (m_u - m_v)uv = \tau_r - d_r r - d_{r|r|}r|r|, \tag{26c}$$

where m_x denotes the diagonal element of the mass-inertia matrix relevant to the degree of freedom x and d_x and $d_{x|x|}$ its linear and quadratic drag coefficients. In this paper, we will use the parameters for the dynamics obtained for the Medusa AUV in Ferreira (2015).

For velocity control, since it is assumed that there are no speed sensors on board, the forward thrust is assigned as feedforward control as follows

$$\tau_u = d_u \|u_k^i\| - d_{u|u|} \|u_k^i\|^2. \tag{27}$$

Since we assume that the yaw is measured we can set the yaw moment as a feedback control of the following form

$$\tau_r = k_r \left(\text{ang}\left(u_k^i\right) - \theta \right). \tag{28}$$

The velocity reference u_k^i is provided by the formation controller described in the next section.

2.4. Formation control

Regarding the formation control, to drive the relative locations $p_k^i - p_k^j$ to their assigned value d_{ij}, the formation controller sets the reference velocities. In Ren and Beard (2008) an adaptation to the conventional continuous time consensus technique for single integrators is suggested to accomplish this. The reference velocities are determined using

$$u_k^i = -K_p \sum_{j=1}^{N} (\hat{p}_k^i - \hat{p}_k^j - d_{ij}), \tag{29}$$

where $\hat{p}_k^i$ is the position estimate of vehicle i.

Position estimates are obtained using the cooperative navigation algorithm. Because our cooperative navigation algorithm uses the control inputs applied to the remaining vehicles, it is necessary to obtain those values as to avoid the traditional assumption of uncertain control actions. Therefore, each vehicle maintains a set-based observer to emulate the individual observers that are running in each vehicle. As a consequence, since the centre of the sets are typically used as estimates of the true position, the values used for all nodes can be retrieved by performing the same calculation. Thus, the cooperative navigation is distributed but all the nodes perform the same updates at some point in time once all the estimates and measurements have been sent.

2.5. Cooperative navigation

In this section, we describe the set-based cooperative navigation algorithm in the context of underwater formation control and drone fire detection.

2.5.1. Cooperative navigation and control architecture

The configuration considered for the cooperative navigation and control algorithm involves steering a vehicle using control inputs u_k^i produced by a cooperative controller that performs a formation control task and receives state estimates X_k of every vehicle in the formation. The latter is calculated by a cooperative observer who gets local measurements and information from the other vehicles. We assume a TDMA communication scheme where each vehicle broadcasts information in its allocated time slot. In particular, we consider that at time $t = k\Delta_t$ vehicle $i = \text{mod}(k-1, N) + 1$ broadcasts data to the other vehicles. The local measurement consists of the range and bearing measurements of the vehicle that sent the information during that time interval. We assume that there are no packet losses and that the messages are transmitted to every vehicle on the network. Because of this, if we compute another estimate

$$\hat{x}_k := \begin{bmatrix} \hat{p}_k^1 \\ \vdots \\ \hat{p}_k^N \end{bmatrix} \tag{30}$$

which only changes based on the data that is sent and received, we design an observer that is synchronized with all the other vehicles, meaning that the state estimates $\hat{x}_k$, and therefore $\hat{p}_k^i$ for all $i \in \{1, \dots, N\}$, is the same on all the vehicles. This synchronized observer is necessary for the cooperative navigation algorithm because it requires the knowledge of the control inputs u_k^i used on the other vehicles as their state estimations are known.

2.5.2. Cooperative underwater navigation

The states of every vehicle in the formation X_k are estimated at each vehicle using the cooperative navigation algorithm. The approach taken into consideration in this study entails updating just the estimations of the vehicle's own state, x_k^i given the predicted states of all the vehicles after receiving the local measurements. Then, the vehicles transmit a reduced version of the updated state to the other vehicles. This approach is illustrated in Algorithm 1, where the method to obtain the CCG $Y(y_k^{i,j})$ is given in Section 2.1.3, $C^i := e^{i\intercal} \otimes I_2$ and $C^{i,j} := (e^i - e^j)^\intercal \otimes I_2$.

Input: Previous state estimate X_{k-1}; control inputs of all vehicles u_{k-1}
Output: State estimate X_k, control input u_k
Predict:
$X_k = AX_{k-1} + Bu_{k-1}$
$\hat{x}_k = A\hat{x}_{k-1} + Bu_{k-1}$
if *Broadcast information* ($i = \text{mod}(k-1, N) + 1$) **then**
  Broadcast $X_k^{send} = \text{ord\_red}(C^i X_k, n_{transmit})$
  $\hat{p}_k^i = \text{centre}\left(X_k^{send}\right)$
end
else if *Receive information from* j ($j = \text{mod}(k-1, N) + 1$) **then**
  Update:
  $X_k = X_k \cap_{C^j} X_k^{receive} \cap_{C^{i,j}} Y(y_k^{i,j})$
  $\hat{p}_k^j = \text{centre}\left(X_k^{receive}\right)$
end
if $i = 1$ **then**
  Update with GPS position:
  $X_k = X_k \cap_{C^1} \left(p_u I_2, y_k^p, \mathbf{0}, 0, \{\mathcal{B}_2\}\right)$
end
if $\text{order}(X_k) \geq 2n_{max}$ **then**
  $X_k = \text{ord\_red}(X_k, n_{max})$
end
for $j = \{1, \dots, N\}$ **do**
  $u_k^j = -K_p \sum_{l=1}^{N} (\hat{p}_k^j - \hat{p}_k^l - d_{jl})$
end
return X_k

Algorithm 1: Cooperative navigation algorithm

On Algorithm 1 the operator $\text{order}(X)$ gives the order of the CCG X, that is supposing that $X := (G_X, c_X, A_X, b_X, \mathfrak{C}_X)$, $\text{order}(X) = \lceil \frac{n_g}{n} \rceil$ where n_g is the number of columns in G_X and n is the number of rows. The operator $Y(\cdot)$ corresponds to the measurement set and is defined by (10). The operator $\text{ord\_red}(X, n_{max})$ represents an order reduction operator described in Silvestre (2023), that is, it outputs a CCG with an order lower or equal to $2n_{max}$ that overbounds the original set. It amounts to find an enclosing polygon with $2n_{max} - 2$ sides. Finally, the operator $\text{centre}(X)$ gives the centre of the CCG X. In order to account for limited communication bandwidth, at each time a reduced order set is transmitted which corresponds to send matrices $G \in \mathbb{R}^{2 \times 2n_{transmit}}$, $c \in \mathbb{R}^2$, $A \in \mathbb{R}^{2(n_{transmit}-1) \times 2}$, and $b \in \mathbb{R}^{2(n_{transmit}-1)}$. This results in communicating $6n_{transmit} - 4$ numbers.

Remark 1. We assume that each AUV can estimate the positions of all vehicles using both their own measurements and data communicated from others. This distributed estimation approach ensures that the fleet's overall state can be reconstructed over time through shared information, thereby achieving global observability. Thus, while local measurements alone are insufficient for complete state estimation, the collective measurements from all vehicles, facilitated by inter-vehicle communication, enable accurate estimation of the positions of all AUVs in the fleet.

Remark 2. There are many estimation-based target tracking algorithms that can be used for observing the state of a fleet of vehicles. For example, Doostmohammadian, Taghieh and Zarrabi (2021) discusses distributed estimation scenarios in which computation and data processing are spread across the sensor network, with estimation localized under global observability assumptions. Another relevant work (Doostmohammadian, Khan, Pirani & Charalambous, 2021) addresses delay-tolerant strategies to manage possible time delays in communication networks. These scenarios involve distributed observability of a target vehicle via a network of communicating sensors on vehicles, distributing both communication and computation over a multi-agent sensor network.

Unlike traditional consensus-based estimators, our method assumes that disturbances and measurement noise are within known sets, providing an estimate of the possible locations of the state. This approach improves upon existing methods by effectively handling bounded uncertainties, reducing communication and computation load on agents, and relaxing the observability assumptions.

Remark 3. The most computationally intensive operation of Algorithm 1 is the order reduction algorithm. This operation has a polynomial complexity.

2.5.3. Drone fire detection algorithm

The same principle of the distributed observer proposed here can also be applied to the problem of fire detection. For fire detection, the cooperative navigation algorithm is given by Algorithm 2.

Input: Previous state estimate X_{k-1}; control inputs of all vehicles u_{k-1}
Output: State estimate X_k, control input u_k
Predict:
$X_k = AX_{k-1} + Bu_{k-1}$
$\hat{x}_k = A\hat{x}_{k-1} + Bu_{k-1}$
if *Broadcast information* $(i = \mathrm{mod}(k-1, N) + 1)$ **then**
  Broadcast $X_k^{send} = \mathrm{ord\_red}(C^i X_k, n_{transmit})$
  $\hat{p}_k^i = \mathrm{centre}\left(X_k^{send}\right)$
end
else if *Receive information from* j $(j = \mathrm{mod}(k-1, N) + 1)$ **then**
  Update:
  $X_k = X_k \cap_{C^j} X_k^{receive} \cap_{C^{i,j}} \left(p_u I_2, y_k^{i,j}, \mathbf{0}, 0, \{\mathcal{B}_2\}\right) \cap_{C^1} Y(y_k^i)$
  $\hat{p}_k^j = \mathrm{centre}\left(X_k^{receive}\right)$
end
if $\mathrm{order}(X_k) \geq 2n_{max}$ **then**
  $X_k = \mathrm{ord\_red}(X_k, n_{max})$
end
for $j = \{1, \ldots, N\}$ **do**
  $u_k^j = -K_p \sum_{l=1}^{N} (\hat{p}_k^j - \hat{p}_k^l - d_{jl})$
end
return X_k

Algorithm 2: Cooperative fire detection algorithm

2.6. Overall system

The integration of all algorithms within the system is illustrated in Fig. 4. In the system, each vehicle i receives a control input u_k^i from the formation control law, which serves as a velocity reference for the velocity tracker. The vehicle then provides sensor measurements, which can be either the relative position $y_k^{i,j}$ or, in certain cases (such as surface vehicles in underwater scenarios or all vehicles in the drone fire detection case), the absolute position y_k^i or y_k^p.

The Cooperative Navigation algorithm operates by receiving a set $X_k^{receive}$ from a vehicle in the network at each time step. When it is a vehicle's turn, it transmits a set X_k^{send} to all other vehicles in the network. The primary output of the Cooperative Navigation algorithm is a set X_k that encompasses all possible positions of the vehicles in the fleet, along with a vector $\hat{x}_k$ containing position estimates of all vehicles, which are then utilized by the formation controller.

3. Results

3.1. AUV formation control

We performed numerical simulations of a fleet of four vehicles to demonstrate the viability of this approach. Vehicle 1 is equipped with a GPS and one vehicle transmits data every $\Delta_t = 0.5s$, so the model has a discretization step of $0.5s$. In order to achieve the appropriate relative locations $d_{1j} := p^1 - p^j$, the follower vehicles must travel in a triangle shape with the leader vehicle as its centre, where the values d_{1j} are given by

$$d_{12} = \begin{bmatrix} 5 \\ 0 \end{bmatrix} \; d_{13} = \begin{bmatrix} -5 \\ -5 \end{bmatrix} \; d_{14} = \begin{bmatrix} -5 \\ 5 \end{bmatrix}. \tag{31}$$

The desired order of the CCG is $n_{max} = 100$, the order of the transmitted set is $n_{transmit} = 4$, the formation control gain is $K_p = 0.1$, the range and bearing measurement parameters are $r_u = r_l = 0.01$ and $b_u = b_l = 10°$, the magnitude of the disturbance is given by $\|w_k\|_\infty \leq 0.1$ and the magnitude of the GPS noise is given by $\|v_k^p\|_\infty \leq 0.1$. The results of the observer are given in Fig. 5, where the red sets correspond to the own position estimates of the vehicles with the proposed method, while the blue sets are calculated using constrained zonotopes (CZs), as shown in Fig. 2. The black lines correspond to the vehicle's trajectories. It can be seen that the method using CCGs is less conservative.

The evolution of the area of the state sets for vehicles 2 to 4 is illustrated in Figs. 6 through 8. For vehicle 1, the state estimate remains consistently small due to its GPS capability and does not benefit significantly from the cooperative navigation algorithm.

From Figs. 6 to 8 one can observe that there is an advantage to a CCG representation in the cooperative navigation algorithm. Note that before 2.5 s, the vehicles had not yet received information from the surface vehicle and therefore their state estimate set is very large.

3.2. Drone formation control

Similarly, we performed numerical simulations of a group of three drones to demonstrate the viability of the drone formation control approach. The drones must stay in a triangle shape with the fire source at its centre, where the values d_{1j} are given by

$$d_{12} = \begin{bmatrix} 10 \\ 5 \end{bmatrix} \; d_{13} = \begin{bmatrix} 10 \\ -5 \end{bmatrix}. \tag{32}$$

As in the previous example the model has a discretization step of $0.5s$. The desired order of the CCG is $n_{max} = 20$, the order of the transmitted set is $n_{transmit} = 4$, the formation control gain is $K_p = 0.1$, the range and bearing measurement parameters are $r_u = r_l = 0.01$ and $b_u = b_l = 30°$, and the magnitude of the disturbance is given by $\|w_k\|_\infty \leq 0.1$. The results of the observer are given in Fig. 9, where it is observed that the method using CCGs is less conservative. We can also observe the great benefit of the cooperative navigation algorithm compared to a non-cooperative algorithm where the vehicles do not communicate given by the sets in green. Note that the cooperative navigation algorithm estimates the relative position of the vehicles relative to the fire source, and therefore despite the high accuracy of the GPS the large area of the set is due to the low precision of the camera sensors.

The evolution of the area of the state sets for vehicles 1 to 3 is given in Figs. 10 to 12.

From Figs. 10 to 11 one can observe that there is a clear advantage to a CCG representation in the cooperative navigation algorithm.

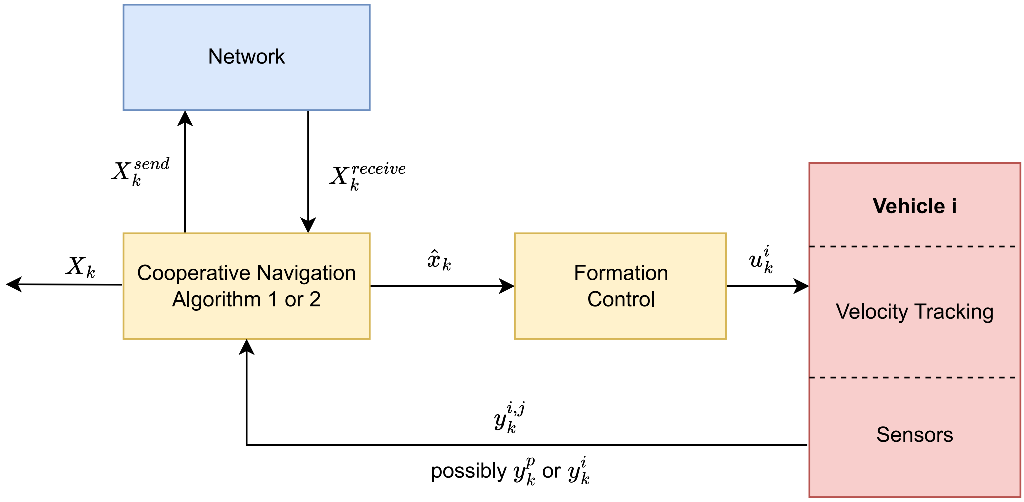

Fig. 4. Overall system diagram.

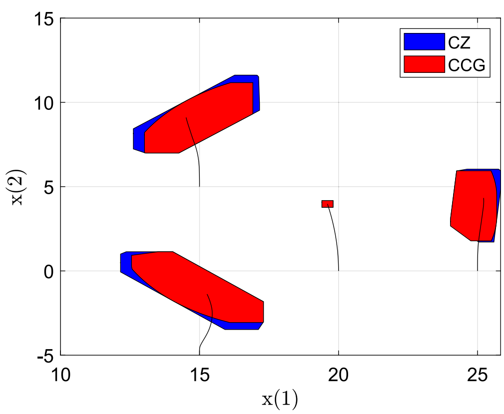

Fig. 5. Results of the observer at iteration $k = 40$.

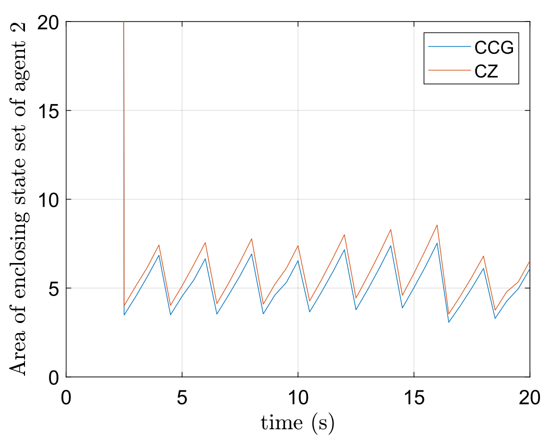

Fig. 6. Evolution of the area of the state estimate set on agent 2.

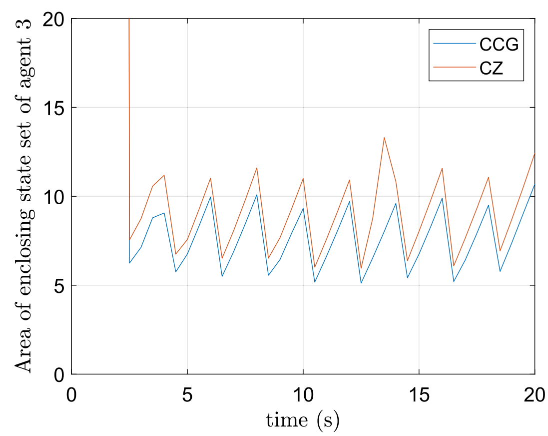

Fig. 7. Evolution of the area of the state estimate set on agent 3.

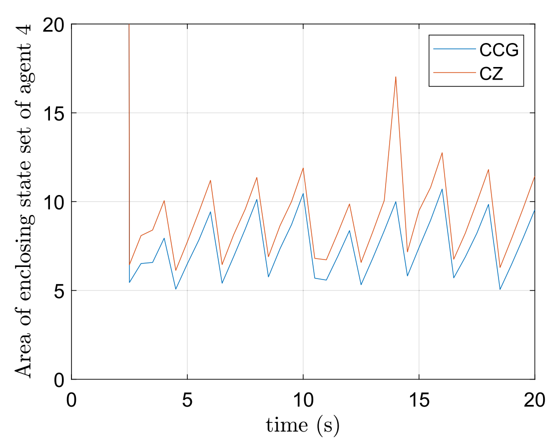

Fig. 8. Evolution of the area of the state estimate set on agent 4.

4. Conclusions

In this paper, we provide a method for set-based cooperative vehicle fleet navigation with low communication bandwidth. With the recommended method, only the distances and bearings to other vehicles are measured. Based on measurements and information transmitted among the vehicles, each one of the vehicles must estimate its own position and the positions of the other vehicles. The vehicles broadcast a reduced set to the other vehicles that contain the state estimation in an effort to minimize the amount of information sent. The obtained estimates are worst-case bounds for the true position, which is important in certain

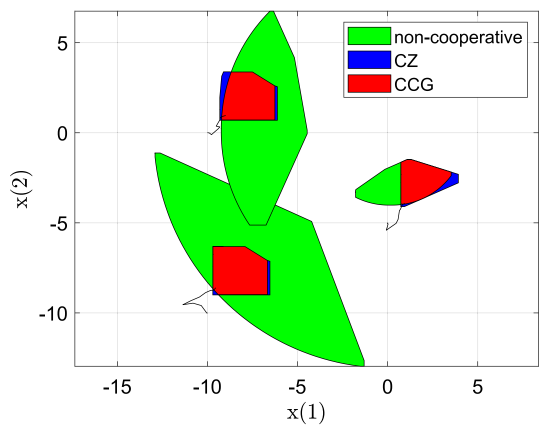

Fig. 9. Results of the observer for fire detection at iteration $k = 10$.

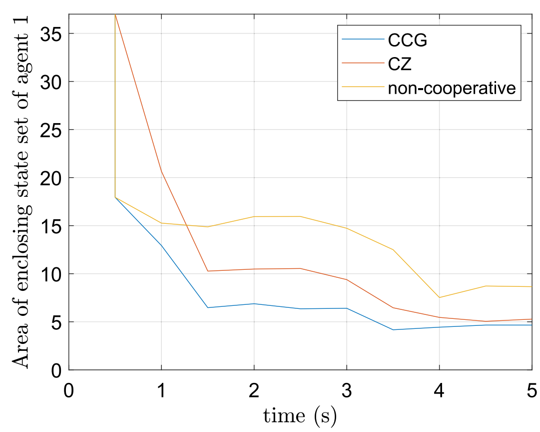

Fig. 10. Evolution of the area of the state estimate set on agent 1.

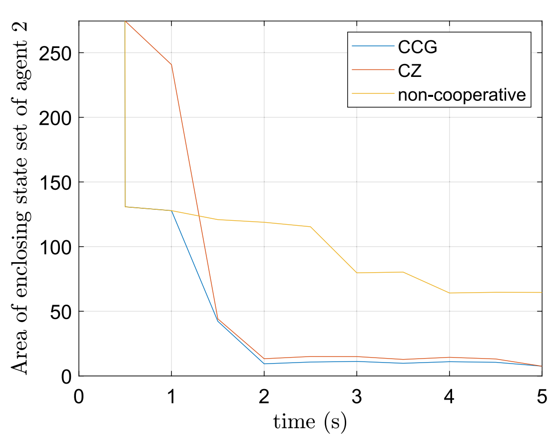

Fig. 11. Evolution of the area of the state estimate set on agent 2.

applications such as collision avoidance. Taking into consideration the positions of all the agents, the calculated sets are subsequently used in vehicle control algorithms. We demonstrate the use of this strategy for the problems of fire detection with numerous UAVs and cooperative underwater vehicle navigation using numerical simulations.

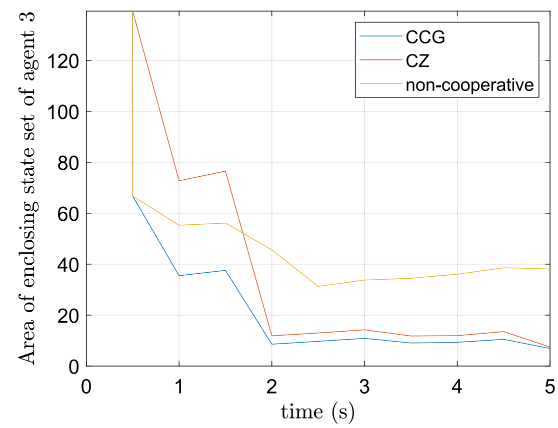

Fig. 12. Evolution of the area of the state estimate set on agent 3.

CRediT authorship contribution statement

Francisco Rego: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Daniel Silvestre:** Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Funding

This work was partially supported 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 UIDB/50009/2020, through project PCIF/MPG/0156/2019 FirePuma and through COPELABS, University Lusófona project UIDB/04111/2020.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Daniel Silvestre reports financial support was provided by Foundation for Science and Technology. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alamo, T., Bravo, J., & Camacho, E. (2005). Guaranteed state estimation by zonotopes. *Automatica*, *41*(6), 1035–1043. http://dx.doi.org/10.1016/j.automatica.2004.12.008.

Alanwar, A., Rath, J. J., Said, H., Johansson, K. H., & Althoff, M. (2020). Distributed set-based observers using diffusion strategy. arXiv preprint arXiv:2003.10347.

Althoff, M., & Rath, J. J. (2021). Comparison of guaranteed state estimators for linear time-invariant systems. *Automatica*, *130*, Article 109662.

Bako, L., Ndiaye, S., & Blanco, E. (2022). An interval-valued recursive estimation framework for linearly parameterized systems. *Systems & Control Letters*, *168*, Article 105345. http://dx.doi.org/10.1016/j.sysconle.2022.105345.

Basu, H., & Yoon, S. Y. (2019). Distributed state estimation by a network of observers under communication and measurement delays. *Systems & Control Letters*, *133*, Article 104554. http://dx.doi.org/10.1016/j.sysconle.2019.104554.

Battistelli, G., & Chisci, L. (2016). Stability of consensus extended Kalman filter for distributed state estimation. *Automatica*, *68*, 169–178.

Benson, B., Li, Y., Kastner, R., Faunce, B., Domond, K., Kimball, D., et al. (2010). *Design of a low-cost, underwater acoustic modem for short-range sensor networks*. IEEE.

Chernousko, F. (2005). Ellipsoidal state estimation for dynamical systems. *Nonlinear Analysis. Theory, Methods & Applications*, *63*(5–7), 872–879. http://dx.doi.org/10.1016/j.na.2005.01.009.

Combastel, C. (2015). Zonotopes and Kalman observers: Gain optimality under distinct uncertainty paradigms and robust convergence. *Automatica*, *55*, 265–273.

Combastel, C. (2016). 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.

Combastel, C., & Zolghadri, A. (2018). FDI in cyber physical systems: A distributed zonotopic and Gaussian Kalman filter with bit-level reduction. *IFAC-PapersOnLine*, *51*(24), 776–783.

Dong, X., Yu, B., Shi, Z., & Zhong, Y. (2014). Time-varying formation control for unmanned aerial vehicles: Theories and applications. *IEEE Transactions on Control Systems Technology*, *23*(1), 340–348.

Doostmohammadian, M., Khan, U. A., Pirani, M., & Charalambous, T. (2021). Consensus-based distributed estimation in the presence of heterogeneous, time-invariant delays. *IEEE Control Systems Letters*, *6*, 1598–1603.

Doostmohammadian, M., Taghieh, A., & Zarrabi, H. (2021). Distributed estimation approach for tracking a mobile target via formation of UAVs. *IEEE Transactions on Automation Science and Engineering*, *19*(4), 3765–3776.

Ferreira, J. G. S. B. (2015). *Nonlinear system identification of autonomous marine vehicles* (Master's thesis), Instituto Superior Técnico.

García, R. A., Orihuela, L., Millán, P., Rubio, F. R., & Ortega, M. G. (2020). Guaranteed estimation and distributed control of vehicle formations. *International Journal of Control*, *93*(11), 2729–2742.

García, R. A., Rubio, F. R., Orihuela, L., Millán, P., & Ortega, M. G. (2017). Observadores distribuidos garantistas para sistemas en red. *Revista Iberoamericana de Automática e Informática Industrial*, *14*(3), 256–267.

González-García, J., Gómez-Espinosa, A., Cuan-Urquizo, E., García-Valdovinos, L. G., Salgado-Jiménez, T., & Escobedo Cabello, J. A. (2020). Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions. *Applied Sciences*, *10*(4), 1256.

Kieffer, M. (2009). Distributed bounded-error state estimation. *IFAC Proceedings Volumes*, *42*(10), 360–365.

Kurzhanski, A., & Varaiya, P. (2000). Ellipsoidal techniques for reachability analysis: internal approximation. *Systems & Control Letters*, *41*(3), 201–211. http://dx.doi.org/10.1016/S0167-6911(00)00059-1.

Li, H., Xie, P., & Yan, W. (2016). Receding horizon formation tracking control of constrained underactuated autonomous underwater vehicles. *IEEE Transactions on Industrial Electronics*, *64*(6), 5004–5013.

Mitra, A., & Sundaram, S. (2018). Distributed observers for LTI systems. *IEEE Transactions on Automatic Control*, *63*(11), 3689–3704.

Olfati-Saber, R. (2007). Distributed Kalman filtering for sensor networks. In *2007 46th IEEE conference on decision and control* (pp. 5492–5498). IEEE.

Orihuela, L., Millán, P., Roshany-Yamchi, S., & García, R. A. (2018). Negotiated distributed estimation with guaranteed performance for bandwidth-limited situations. *Automatica*, *87*, 94–102.

Orihuela, L., Roshany-Yamchi, S., García, R. A., & Millán, P. (2017). Distributed set-membership observers for interconnected multi-rate systems. *Automatica*, *85*, 221–226.

Pourasghar, M., Puig, V., & Ocampo-Martinez, C. (2019). Interval observer versus set-membership approaches for fault detection in uncertain systems using zonotopes. *International Journal of Robust and Nonlinear Control*, *29*(10), 2819–2843.

Rego, F. F. (2023). Distributed observers for LTV systems: A distributed constructibility gramian based approach. *Automatica*, *155*, Article 111117.

Rego, F. F., Pascoal, A. M., Aguiar, A. P., & Jones, C. N. (2019). Distributed state estimation for discrete-time linear time invariant systems: A survey. *Annual Reviews in Control*, *48*, 36–56.

Ren, W., & Beard, R. W. (2008). *Distributed consensus in multi-vehicle cooperative control*: *vol. 27*, Springer.

Scott, J. K., Raimondo, D. M., Marseglia, G. R., & Braatz, R. D. (2016). Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, *69*, 126–136.

Silvestre, D. (2022a). Constrained convex generators: A tool suitable for set-based estimation with range and bearing measurements. *IEEE Control Systems Letters*, *6*, 1610–1615. http://dx.doi.org/10.1109/LCSYS.2021.3129729.

Silvestre, D. (2022b). Set-valued estimators for Uncertain Linear Parameter-Varying systems. *Systems & Control Letters*, *166*, Article 105311. http://dx.doi.org/10.1016/j.sysconle.2022.105311, URL: https://www.sciencedirect.com/science/article/pii/S0167691122001141.

Silvestre, D. (2023). Exact set-valued estimation using constrained convex generators for uncertain linear systems. *IFAC-PapersOnLine*, *56*(2), 9461–9466. http://dx.doi.org/10.1016/j.ifacol.2023.10.241, 22nd IFAC World Congress.

Silvestre, D., Hespanha, J. P., & Silvestre, C. (2019). Broadcast and gossip stochastic average consensus algorithms in directed topologies. *IEEE Transactions on Control of Network Systems*, *6*(2), 474–486. http://dx.doi.org/10.1109/TCNS.2018.2839341.

Silvestre, D., Rosa, P., Hespanha, J. P., & Silvestre, C. (2014). Finite-time average consensus in a byzantine environment using set-valued observers. In *2014 American control conference* (pp. 3023–3028). IEEE.

Silvestre, D., Rosa, P., Hespanha, J. P., & Silvestre, C. (2015). Distributed fault detection using relative information in linear multi-agent networks. *IFAC-PapersOnLine*, *48*(21), 446–451.

Silvestre, D., Rosa, P., Hespanha, J. P., & Silvestre, C. (2017). Stochastic and deterministic fault detection for randomized gossip algorithms. *Automatica*, *78*, 46–60. http://dx.doi.org/10.1016/j.automatica.2016.12.011, URL: https://www.sciencedirect.com/science/article/pii/S0005109816305192.

Wang, Y., Alamo, T., Puig, V., & Cembrano, G. (2018). A distributed set-membership approach based on zonotopes for interconnected systems. In *2018 ieee conference on decision and control* (pp. 668–673). IEEE.

Wang, S., & Ren, W. (2018). 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. http://dx.doi.org/10.1109/TCST.2017.2715849.