

## Distributed Exploration Algorithm for GPS-denied Areas based on Flocking Rules

Rafael Ribeiro<sup>a</sup>, Francisco Santos<sup>a</sup>, Daniel Silvestre<sup>a,b,c</sup> and Carlos Silvestre<sup>a,d</sup>

<sup>a</sup> Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal;

<sup>b</sup> School of Science and Technology from the NOVA University of Lisbon (FCT/UNL), 2829-516 Caparica, Portugal

<sup>c</sup> COPELABS from the Lusófona University

<sup>d</sup> Department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Macau, Macau, China, on leave from Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal.

### ARTICLE HISTORY

Compiled February 6, 2025

### ABSTRACT

This paper addresses the problem of having a Multi-Agent System (MAS) search for areas of higher relative importance as measured by an unknown utility function. In the envisioned scenario, there are inexpensive agents without localization sensors and limited communication capabilities. More expensive nodes serve as fixed towers and forward noisy position and velocity measurements using directional antennae. There is no assumptions on initial network connectivity. By proposing a set of flocking rules and set-membership estimation, the formation drives to a vicinity of nearby local maximum of the function while having theoretical guarantees of no collisions. The performance of the method is evaluated both in Matlab simulations and using the CrazySwarm package under the Robot Operating System (ROS) environment, including cases of moving destinations, obstacles, undesirable zones, and different number of nodes and sizes of the mission plane.

### KEYWORDS

Sections; lists; figures; tables; mathematics; fonts; references; appendices

## 1. Introduction

The problem of rendezvous consists in devising a distributed algorithm to make a set of mobile agents converge to defined areas of the mission plane. This paper focuses on proposing distributed control rules that can rendezvous agents with limited connectivity. We assume a set of identical agents, i.e., having the same communication capabilities, no localization mechanisms, and that must follow a similar control law. Having such a method would mean reduced computational complexity in the system and fewer points-of-failure by avoiding the need for hierarchical protocols or specialized

---

Contacts: R. Ribeiro [rribeiro@isr.tecnico.ulisboa.pt](mailto:rribeiro@isr.tecnico.ulisboa.pt), F. Santos [francisco.velez@tecnico.ulisboa.pt](mailto:francisco.velez@tecnico.ulisboa.pt), D. Silvestre [dsilvestre@isr.tecnico.ulisboa.pt](mailto:dsilvestre@isr.tecnico.ulisboa.pt) and C. Silvestre [csilvestre@umac.mo](mailto:csilvestre@umac.mo).

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.

message exchanges. Furthermore, the method is capable of coping with asynchronous communications and dynamic networks with agents arbitrarily joining and leaving, among other real-world problems.

The envisioned scenario corresponds to a mission to be carried out by low cost autonomous agents in a formation where some of them have additional localization capabilities. Examples range from defense where there is specialized equipment in establishing the communication and localization infrastructure to be used by lightweight drones, to agricultural settings where the payload weight restriction comes from the need to save space and have longer autonomy time to carry the mission (either planting trees, inspecting bugs in crops, etc). All these scenarios can be cast in the framework considered in this paper. There are fixed control towers responsible for measuring the system state and sending it with the use of directional antennae to the mobile agents. The agents movement is asynchronous and triggered upon receiving a tower communication (that can also be received by other nodes that are listening to the medium). These messages contain position and velocity of the agents. Thus, each agent is only equipped with the necessary sensors to guarantee its desired movement is achieved and an antenna to receive the information broadcasts. Through the use of directional communication and decentralized control laws, our proposed algorithm has an improved use of the shared wireless medium and is suited for large networks.

The rendezvous problem in the area of control theory is typically addressed through the use of consensus algorithms, in which each node computes the weighted average of its neighbors' positions and moves towards that position. However, a prevalent assumption among these works is that of a connected network topology (or at least connected over time). Such an assumption cannot be posed in our scenario and a pure-consensus approach would result in multiple consensus sub-groups converging to their own values. Throughout the literature, this problem is solved with the use of linear iterative algorithms that converge to a weighted average of the initial state. Several different scenarios have been addressed for consensus problems: networks with switching topology and time-delays Olfati-Saber and Murray (2004), networks with communication-link failures Fagnani and Zampieri (2009); Patterson, Bamieh, and El Abbadi (2010), networks with stochastic and asymmetric communication Antunes, Silvestre, and Silvestre (2011); Silvestre, Hespanha, and Silvestre (2019), networks with quantized data transmission Carli, Bullo, and Zampieri (2010), and event-triggered D. Dimarogonas and Johansson (2009) and self-triggered D. V. Dimarogonas, Frazzoli, and Johansson (2010) control. Our proposed solution combines traditional consensus techniques with improved flocking behavior, Reynolds (1987), to account for disconnected network topologies and mitigate the number of consensus sub-groups.

Considering imperfect measurements is also critical in rendezvous problems. In Sadikhov, Haddad, Goebel, and Egerstedt (2014), the position of an agent is assumed to be a ball of radius  $r$  that contains the real uncertain position. Contrarily to our scenario, their agents know the network topology and the total number of agents in the system. In Silvestre, Rosa, Hespanha, and Silvestre (2015), an agent uncertain position was generalized to be a convex polytope, and it was shown that estimates for the true position of the neighbors can be accomplished resorting to the use of Set-Valued Observers (SVOs). Moreover, building on the results in Silvestre, Rosa, Hespanha, and Silvestre (2017) and Shamma and Tu (1999), if all sets are polytopes and the dynamics has no uncertainties, polytopes are exact representations. However, we will be using a generalization named Constrained Convex Generators (CCGs) Silvestre (2022a) that extend the type of sets that can be used to have polytopes, ellipsoids, and any shape that results from intersections of those types, which is often the case in the presence

of range measurements. We also remark that a point-based method was a viable alternative given the reduced state space Silvestre (2022b). Given that the agent will employ a guaranteed state estimation, it is a straightforward feature that the system is guaranteed no collisions as long as none of the sets intersect.

The main contributions of this paper are as follows:

- A exploration algorithm to find dynamic rendezvous targets which can handle disconnected network topologies, asynchronous and directional communication, imperfect position measurements, and a utility-defined mission plane, while guaranteeing no collisions between agents and convergence to a single area, provided there exists sufficient node density and a single maximum;
- Extensive simulations illustrating the emergent behavior of the agents in various scenarios corresponding to real-world missions;
- A result stating that our method drives the nodes to a number of clusters no greater than the number of clusters in the initial configuration.

All these contributions are achieved while avoiding the assumptions present in the literature of a known and connected network topology and a known number of total agents in the system, as Sadikhov et al. (2014) and Silvestre et al. (2019). Therefore, the main differentiating factor with other methods in the literature is that the proposed method is able to cope with disconnected topologies, has a decentralized decision based on the constructed estimates by each vehicle and will drive to local maximizers of the utility function. A preliminary version of the proposed algorithm was given in Ribeiro, Silvestre, and Silvestre (2020); Ribeiro, Silvestre, and Silvestre (2021) for the part of achieving consensus in disconnected topologies.

An outline of this paper is as follows. Section 2 presents background material followed by the problem statement. The proposed rendezvous algorithm for a multi-agent system with various targets is described in Section 4. Section 5 presents theoretical convergence results and guarantees of the method, while simulations are provided in Section 6 for other cases of interest. Concluding remarks and directions for future work are presented in Section 7.

## 2. Problem statement

This paper addresses the problem of a MAS composed of an unknown number  $n$  of agents whose principal objective is to rendezvous on multiple dynamic targets in order to accomplish a mission. To avoid the need for hierarchical protocols or specialized message exchanges, we consider the agents to be identical, and therefore, to be constrained to follow the same control law. Additionally, to develop an inexpensive solution, the agents are not equipped with localization sensors, only having odometry sensors to verify the desired movement is executed and the limited communication capabilities necessary to receive the towers broadcasts regarding their state.

### 2.1. Mission plane

Our proposed mission plane is defined by an utility function  $h$  that assigns a value to the agents positions to reflect how desirable they are to the mission objectives. Thus, the agents must drive towards the maximum of  $h$  by only having access to the utility value and gradient at the current position. In this paper, it is assumed that the (possibly nonconvex) function  $h$  is defined as a finite sum of paraboloids.

## 2.2. Towers

In this paper, we are assuming that a small subset of expensive agents will act as communication towers (i.e., will become stationary) in order to provide localization for the larger subset of inexpensive nodes. For convenience, we will refer to these agents as towers to clearly distinguish from the inexpensive nodes and assume that they are stationary for the entirety of the mission. Since towers are equipped with directional antennae, it composes a message comprising the utility value, position and velocity estimates of each agent and transmits it to that node. Therefore, any node in a communication cone can receive eavesdrop the message. We also assume that the tower serves in a round robin fashion each node to update their estimates. In doing so, power is saved in the communication and results in a better use of the shared medium. Notice that for large networks the sent messages in each direction are reduced in size since a full broadcast is avoided.

Due to the absence of localization sensors, the agents rely on the messages from the towers to determine their location and calculate the proposed control law, described in Section 4. These measurements are imperfect, corrupted by noise, meaning that instead of accessing the true state, the towers determine a set-valued estimate  $\mathcal{X}_i(k)$  containing all possible positions for agent  $i$  at time  $k$ , whose centroid  $c_i(k)$  will typically not correspond to the true value  $x_i(k)$ . Similarly, the estimate also comprises in other coordinates the possible values for the velocity  $v_i(k)$ .

The directional communication behavior is illustrated in Figure 1: any two agents  $i$  and  $j$  are considered to be neighbors if they are bounded by the same communication cone. The example represents a cone centered on agent  $i$  with all nodes contained in its projection on the mission plane receiving the broadcast. The existence of agent  $l$  is not acknowledged by agents  $i$  or  $j$ , as this agent is not in the communication strip, resulting in its position not being transmitted.

The set of agents is grouped in  $\mathcal{V}$ , and the undirected links of the form  $(i, j)$  translate into a neighbor relation between  $i$  and  $j$  since both will receive the broadcast with their position and velocity estimates. Consequently, the network topology is modeled as an undirected graph,  $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}(k))$ , where  $\mathcal{E}(k)$  is defined as a set of all undirected links  $(i, j)$  at time instant  $k$ . In general,  $\mathcal{G}_k$  will be disconnected and composed of a finite number of clusters at any given time  $k$ . We do not assume knowledge of the network topology nor can agents determine  $n$ . Agent  $i$  will only have access to its neighbor set  $\mathcal{N}_i(k)$  by listening to the communication channel.

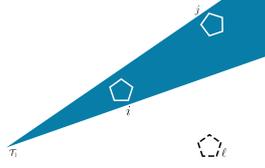
This paper addresses three main issues that appear with the use of directional antennae:

- i) absence of a connected and known network topology;
- ii) limited unidirectional communication;
- iii) position and velocity estimates received at different time instants.

## 3. Constrained Convex Generators overview

In this section, we review for completeness the definition and main set operations when using CCGs that are used in the set-valued estimation of position and velocity for each agent.

**Definition 3.1** (CCG Silvestre (2022a)). A Constrained Convex Generator (CCG)  $\mathcal{Z} \subset \mathbb{R}^n$  is defined by the tuple  $(G, c, A, b, \mathfrak{C})$  with  $G \in \mathbb{R}^{n \times n_g}$ ,  $c \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{n_c \times n_g}$ ,



**Figure 1.** Illustration of a communication tower broadcasting state of the agents inside the projection of the communication cone on the mission plane.

$b \in \mathbb{R}^{n_c}$ , and  $\mathfrak{C} := \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{n_p}\}$  such that:

$$\mathcal{Z} = \{G\xi + c : A\xi = b, \xi \in \mathcal{C}_1 \times \dots \times \mathcal{C}_{n_p}\}. \quad (1)$$

**Definition 3.2** (Set Operations for CCGsSilvestre (2022a)). Consider three CCGs as in Definition 3.1:

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

and a matrix  $R \in \mathbb{R}^{m \times n}$  and a vector  $t \in \mathbb{R}^m$ . The set operations linear map, Minkowski sum and generalized intersection are defined as:

$$R\mathcal{Z} + t = (RG_z, Rc_z + t, A_z, b_z, \mathfrak{C}_z) \quad (2)$$

$$\mathcal{Z} \oplus \mathcal{W} = \left( [G_z \quad G_w], c_z + c_w, \begin{bmatrix} A_z & 0 \\ 0 & A_w \end{bmatrix}, \begin{bmatrix} b_z \\ b_w \end{bmatrix}, \{\mathfrak{C}_z, \mathfrak{C}_w\} \right) \quad (3)$$

$$\mathcal{Z} \cap_R \mathcal{Y} = \left( [G_z \quad 0], c_z, \begin{bmatrix} A_z & 0 \\ 0 & A_y \end{bmatrix}, \begin{bmatrix} b_z \\ b_y \\ c_y - Rc_z \end{bmatrix}, \{\mathfrak{C}_z, \mathfrak{C}_y\} \right). \quad (4)$$

With the aforementioned definition, it is possible to present how to build estimates from a dynamical model of the form:

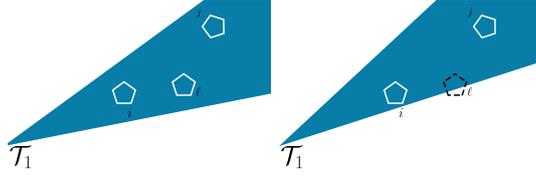
$$x(k+1) = F_k x(k) + B_k u(k) + d(k) y(k) = C_k x(k) + v(k) \quad (5)$$

where  $x(k)$ ,  $u(k)$ ,  $d(k)$  and  $v(k)$  are respectively the state, actuation, disturbance and noise signals at time  $k$ . Matrices  $F_k$ ,  $B_k$  and  $C_k$  are of appropriate dimension and represent the dynamics, how the actuation enters the system and the measurements available. Following the typical two step akin to the Kalman filter, the propagation phase equates to:

$$\mathcal{X}_{\text{prop}}(k+1) = F_k \mathcal{X}(k) + B_k u(k) \oplus \mathcal{D}(k), \quad (6)$$

and the update:

$$\mathcal{X}(k+1) = \mathcal{X}_{\text{prop}}(k+1) \cap_{C_k} (y(k+1) \oplus -\mathcal{V}(k)), \quad (7)$$



**Figure 2.** Strip centered at node  $i$  with  $i$  being neighbors with agents  $j$  and  $l$  (left). Strip centered at node  $j$  with  $i$ 's only neighbor  $j$  (right).

such that  $x(k) \in \mathcal{X}(k)$ ,  $v(k) \in \mathcal{V}(k)$  and  $d(k) \in \mathcal{D}(k)$ .

**Remark 1.** Please notice that even though the set-valued estimation is presented for linear systems, if we adopt a strategy similar to the Extended Kalman filter for nonlinear systems,  $F_k$  corresponds to the derivative of the nonlinear dynamics with respect to the state at the previous center of the set-valued estimate. In a similar light,  $B_k$  would be the derivative with respect to the input and we would have to add a disturbance term similar to  $\mathcal{D}(k)$  to account for the linearization error which corresponds to the set that contains the Lagrange remainder when we truncate the Taylor series expansion after the linear term. The interested reader is pointed to Silvestre (2022a) and the references therein for additional details and references to other possible set representations.

#### 4. Proposed Solution

The proposed algorithm in this paper aims at reducing the number of existing clusters through the use of flocking such that the swarm is coordinated by a fully decentralized algorithm. We improve upon the original flocking rules by adding three additional ones that can be intuitively understood as:

- (1) *Attraction* - a simulated attraction/repulsion force towards neighbor agents based on the difference of utility in their corresponding positions;
- (2) *Utility* - a simulated attraction force towards the direction that maximizes the gradient of the utility function in case it is differentiable at the current location;

We consider the agent movement to be asynchronous since it is triggered by the reception of a new message with the state of all nodes in the cone, i.e., position polytope  $\mathcal{X}_j(k)$ , velocity  $v_i(k) \forall j \in \mathcal{N}_i(k)$  and utility value. Formally, the neighbor set is constructed as:

$$\mathcal{N}_i(k) = \{j \in \mathcal{V} : \angle(\vec{t}_i, \vec{t}_j) \leq \alpha, \mathcal{X}_j(k) \cap \vec{t}_\alpha = \emptyset\} \quad (8)$$

where  $2\alpha$  is the broadcast cone angle,  $\vec{t}_i$  is the vector from the tower's position to agent  $i$ ,  $\vec{t}_j$  is the analogous for each neighbor agent  $j$ , and  $\angle(\vec{t}_i, \vec{t}_j)$  is the angle between them. Intuitively, for an agent to belong to  $\mathcal{N}_i(k)$ , its set-valued position estimate has to be contained in the communication cone of angle  $2\alpha$ . Figures 2(a) and 2(b) illustrate a worst-case scenario by having nodes  $i$  and  $l$  being neighbors in the former, but not in the latter, despite the centroid of the polytope of agent  $l$  being inside the strip.

#### 4.1. Solution Architecture

The proposed strategy can be split into three main components as seen in Figure 3. As the agents receive new data from the towers they start by computing a desired heading vector using the control law defined in Section 4.2. Afterwards, the computed heading is limited as explained in Section 4.3 so as to avoid collisions. Finally, the limited vector is sent to the vehicle local controller.

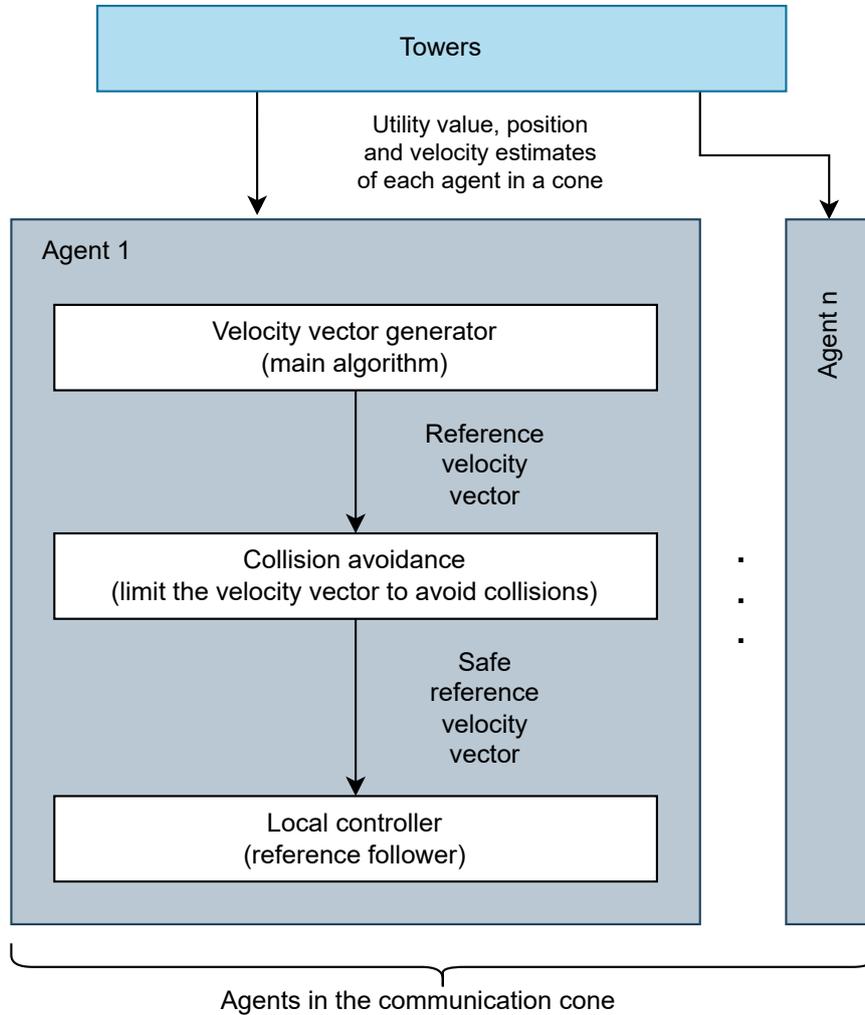


Figure 3. Architecture of the proposed solution.

#### 4.2. Velocity Vector Generator

Upon receiving a message, an agent computes six movement components that constitute the control law: *Separation*, *Cohesion*, *Alignment* - three original flocking rules - and *Attraction*, *Utility* - introduced in this paper to address the issues of the mission definition. This control law is used only to generate the reference velocity vectors which are then tracked by a low level controller (usually a Proportional, Integral and Derivative (PID) controller onboard the vehicle).

#### 4.2.1. Separation component

Responsible for maintaining the agents at a minimum distance: to increase the occupied area of the flock and as a simple collision avoidance method. This component,  $\mathcal{U}_i^s(k)$ , is calculated by:

$$\mathcal{U}_i^s(k) = \frac{1}{|\mathcal{N}_{i,d}(k)|} \sum_{j \in \mathcal{N}_{i,d}(k)} (c_i(k) - c_j(k)) \quad (9)$$

where  $\mathcal{N}_{i,d}(k)$  is the set of agent  $i$ 's neighbors within  $d$  distance units.

#### 4.2.2. Cohesion component

Responsible for preventing fragmentation of a cluster by maintaining the agents within a maximum distance, and is computed by:

$$\mathcal{U}_i^c(k) = \frac{1}{|\mathcal{N}_i(k)|} \sum_{j \in \mathcal{N}_i(k)} (c_j(k) - c_i(k)) \quad (10)$$

In *Separation*, only neighbors closer than  $d$  are considered. This is justified by separating from distant agents being counterproductive to the mission objective of rendezvousing. As *Cohesion* aligns with this objective, all neighbors are considered.

#### 4.2.3. Alignment component

The final original flocking rule, *Alignment*, minimizes the difference in all agents' headings, to maintain the agents moving in the same direction.  $\mathcal{U}_i^a(k)$  is computed with:

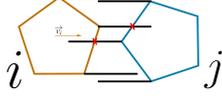
$$\mathcal{U}_i^a(k) = \frac{1}{|\mathcal{N}_i(k)|} \sum_{j \in \mathcal{N}_i(k)} (v_j(k) - v_i(k)) \quad (11)$$

#### 4.2.4. Attraction component

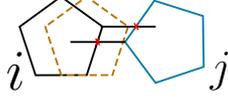
The agents' objective is to rendezvous to the most desirable rendezvous target its cluster has found. The *Attraction* movement component simulates an attraction/repulsion force towards each neighbor, based on the relative difference of utility values in their positions. These forces drive the agent closer to agents in higher-utility positions and away from agents in less desirable areas.  $\mathcal{U}_i^{attr}(k)$  is defined by:

$$\mathcal{U}_i^{attr}(k) = \frac{1}{|\mathcal{N}_i(k)|} \sum_{j \in \mathcal{N}_i(k)} (h(c_j(k)) - h(c_i(k)))(c_j(k) - c_i(k)) \quad (12)$$

where  $h_i$  and  $h_j$  are the utility values at agent  $i$  and each of its neighbor centroid positions, respectively.



**Figure 4.** Rays being cast from both the moving agent’s polytope and an obstacle-agent’s polytope. The red Xs represent the first intersection point between a ray and the opposite polytope.



**Figure 5.** Example of the allowed movement for an agent that would otherwise collide.

#### 4.2.5. Utility component

The second additional component, *Utility*, is also motivated by the objective of maximizing the utility function, by following its gradient. For a non-differentiable  $h$ , it is possible the use of subgradients or assign a zero value.  $\mathcal{U}_i^u(k)$  is calculated with:

$$\mathcal{U}_i^u(k) = \nabla h(c_i(k)) \quad (13)$$

#### 4.2.6. Final control law

The overall control law is given by:

$$u_i(k) = \theta \cdot \mathcal{U}_i^s(k) + \beta \cdot \mathcal{U}_i^c(k) + \gamma \cdot \mathcal{U}_i^a(k) + \delta \cdot \mathcal{U}_i^{attr}(k) + \epsilon \cdot \mathcal{U}_i^u(k) \quad (14)$$

where all the component vectors are normalized, and all weights  $\theta, \beta, \gamma, \delta, \epsilon \in \mathbb{R}^+$ . We remark that tuning these parameters must be based on the relative importance of each of the tasks for the overall mission. However, (14) only defines a direction that the agent must follow since its magnitude will be selected by the proposed collision avoidance method.

### 4.3. Collision Avoidance Strategy and Local Controller

Given that the real position of an agent  $i$ ,  $x_i$ , is contained in its set-valued estimate  $\mathcal{X}_i$  updated by the Set-Valued Observers framework, a direct consequence is that a safe maneuver of the nodes can be achieved by moving such that the polytopes do not intersect. Intuitively, if we are moving the two convex polytopes, they will first intersect with one of their vertices. Consequently, our proposed collision avoidance strategy utilizes ray tracing and line intersection checks to detect collisions. The acceleration vector can be appropriately set such that the magnitude of the velocity vector makes the agent move the maximum allowed distance without colliding.

Elaborating on the technique, from each vertex of the moving agent polytope,  $\mathcal{X}_i$ , we cast rays in the direction of the desired velocity with the length of the maximum distance an agent is allowed to move per discrete update. For each of the rays, we verify if it intersects with the edges of the obstacle-agent polytope,  $\mathcal{X}_j$ . Concurrently, rays are cast from the obstacle-agent’s polytope, in the opposite direction of the desired velocity, with intersection checks against the edges of  $\mathcal{X}_i$ . An illustration of the collision

detection method with two polytopes is provided in Figure 4. Notice that, only casting from the moving agent polytope could result in undetected collisions. Our method consists in executing this pair-wise operation for each neighbor agent  $j \in \mathcal{N}_i$ , as to obtain the most imminent collision point. Figure 5 illustrates the allowed movement distance for agent  $i$ , with agent  $j$  being the cause of the first collision point.

Following the collision avoidance procedure the desired heading vector is sent to the vehicle local controller which acts as a reference follower by using onboard sensors to correct for disturbances when trying to follow the provided reference.

## 5. Convergence Analysis

In this section, we give a convergence result regarding the proposed flocking algorithm by reformulating it as a consensus system with a virtual leader given by the utility values of the players. Given the proposed collision avoidance mechanism, the value  $\theta$  accounting for the weight of the *Separation* velocity vector can be negligible as the nodes will not collide. Moreover, given that positions are corrupted by noise, the theoretical results have to be derived for the centroids of the polytopes. This means that not considering collisions in our analysis and proving convergence translates to convergence to a ball around the target that is determined by the noise magnitude and the number of nodes (since when agents cannot collide they will prevent others from entering within their set-valued estimate for the position).

Given the above considerations, we now present the main theorem of this paper where convergence is proved when disregarding the *Separation* component and the collision avoidance method and where the symbol  $\otimes$  will be used to denote the Kronecker product.

**Theorem 5.1.** *Consider  $n$  nodes with the centroids of  $\mathcal{X}_i(k)$  given by  $c_i(k), \forall i \leq n$  following the algorithm in (14) with a sufficiently small  $\epsilon$  gradient step, in a mission characterized by a concave utility function  $h$ . Then, the velocity iteration is equivalent to a consensus algorithm perturbed by a virtual leader, i.e.,*

$$v(k+1) = Wv(k) + z^{\text{virtual}}(k) \quad (15)$$

where  $W$  corresponds to a consensus matrix and  $z^{\text{virtual}}$  to an input of the system. Moreover, the position of the nodes evolves as a perturbed gradient ascent algorithm and therefore  $\lim_{k \rightarrow \infty} \|x(k) - \mathbf{1}_{2n} \otimes x^*\|_2 \leq \varphi(\eta)$  where  $x^* := \arg \max h(x)$  and  $\varphi(\eta)$  is some constant dependent on the noise level of the measurements  $\eta$ .

**Proof.** We first write the velocity iteration equation as:

$$v_i(k+1) = v_i(k) + \beta \cdot \mathcal{U}_i^c(k) + \gamma \cdot \mathcal{U}_i^a(k) + \delta \cdot \mathcal{U}_i^{\text{attr}}(k) + \epsilon \cdot \mathcal{U}_i^u(k) \quad (16)$$

$$= (1 - \gamma)v_i(k) + \gamma \frac{1}{\mathcal{N}_i(k)} \sum_{j \in \mathcal{N}_i(k)} v_j(k) + z_i^{\text{virtual}}(k) \quad (17)$$

where  $z_i^{\text{virtual}}(k) = \frac{1}{\mathcal{N}_i(k)} \sum_{j \in \mathcal{N}_i(k)} (\beta + \delta(h(c_j(k)) - h(c_i(k))))(c_j(k) - c_i(k)) + \epsilon \nabla h(c_i(k))$ .

The iteration can be expressed in matrix form as:

$$v(k+1) = \underbrace{W_\gamma v(k)}_{\text{consensus}} + \underbrace{z^{\text{virtual}}(k)}_{\text{virtual leader}}. \quad (18)$$

Notice that the input  $z^{\text{virtual}}(k)$  can only be zero if all nodes are located at the maximum of function  $h$ . Moreover, this vector has bounded norm and converges to zero since function  $h$  is concave. The dynamics in (18) make the velocity vector of each agent track the average of the virtual input since  $\gamma$  is selected small enough and provided that the support graph of  $W_\gamma$  is connected. Since the connectivity does not hold in general, let us consider each cluster such that  $W_\gamma$  satisfies the connectivity condition within the cluster.

The position iteration is given by:

$$x_i(k+1) = x_i(k) + v(k) \quad (19)$$

where  $v(k)$  is tracking the average of the gradient for each agent. This is a gradient ascent iteration that converges to the maximum of a concave function provided an appropriately sufficiently small step size is selected. Due to the existence of noise, the conclusion

$$\lim_{k \rightarrow \infty} \|x(k) - \mathbf{1}_{2n} \otimes x^*\|_2 \leq \varphi(\zeta, \eta) \quad (20)$$

follows since each cluster can be viewed as independent gradient ascents, where  $\varphi(\eta)$  is the maximum norm of the noise.  $\square$

## 6. Simulation Results

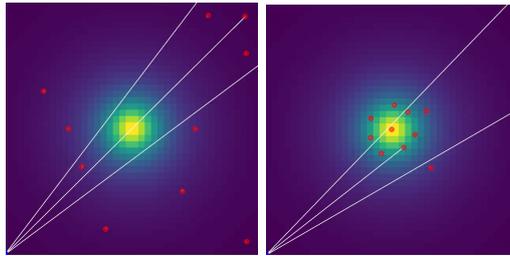
In this section, we present simulation results for various scenarios bearing resemblance with real-world applications and missions. Two methods were used to obtain the simulation results: *Matlab* simulations with a simple double integrator dynamic (using a proportional controller as a local controller, see Figure 3) and no noise were used to validate the algorithm in ideal conditions and to run longer simulations given its lower computational requirements; More realistic simulations provided by the *Crazyswarm* package using *Crazyflie 2.0* vehicles under the *ROS* environment were used as this package uses more realistic dynamics for the drones and aims at validating the algorithm in a more realistic setting. Moreover, the *Crazyswarm* package makes use of the *Crazyflie* firmware bindings which uses a Cascaded PID controller as a local controller (Figure 3).

In the *Matlab* simulations the following setups were used:  $n = 10$  in a  $100 \times 100$  field, and 30 agents in a  $200 \times 200$  mission plane. Simulations with the latter will be referred to be in a large-network environment for brevity. In the *Crazyswarm* simulations the following setups were used:  $n = 10$  in a  $20m \times 20m$  field, and 30 agents in a  $100m \times 100m$  mission plane, along with a simulation time of 180s. Also in the *Crazyswarm* simulations the vehicles are depicted with the set-valued estimate of their position as a red circle around it and the implementation of all the collision avoidance techniques was made using circles instead of polytopes as a representation of the agents. The radius of these circles is based on the size of the vehicle itself as well as on the variance of

the noise in the positioning system. Finally, the following control values were used:  $\theta = 0.07, \beta = .01, \gamma = 0.6, \delta = 0.08, \epsilon = 1.0, \zeta = 1.0, \eta = 0.01$  and in all scenarios, the communication towers angle was 16 degrees. For brevity only the *Crazyswarm* simulations are shown for the scenarios in the smaller map and only the *Matlab* results are shown the larger map. However, the simulation figures and videos of all the simulations are available on the GitHub link [https://github.com/hardtekpt/pdfc\\_fr\\_videos](https://github.com/hardtekpt/pdfc_fr_videos). The scenarios presented are:

- (1) Single maximum utility function to account for rendezvous missions:
  - (a) Static rendezvous point;
  - (b) Moving rendezvous point;
- (2) Single minimum utility function to illustrate escape missions:
  - (a) Static minimum point;
  - (b) Moving minimum point;
- (3) Multiple maxima and minima to depict the case of conflicting objectives:
  - (a) Static rendezvous areas;
  - (b) Static rendezvous areas with a large network;
  - (c) Static rendezvous areas and illegal zones;
  - (d) Static rendezvous areas and illegal zones with a large network;
  - (e) Rendezvous areas with dynamic utility values;
  - (f) Rendezvous areas with dynamic utility values in a mission plane with illegal zones;
  - (g) Random creation and destruction of rendezvous and minimum points.

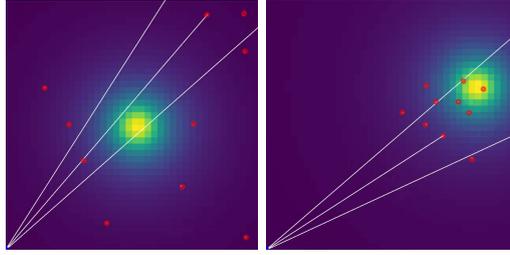
Figures 6(a) and 6(b) illustrate the initial and final configurations of the agents in the simulation type 1a - single static rendezvous point - which is at the center of the mission plane, represented in yellow. This first example shows that convergence can be achieved in a simple mission plane with a single communications tower, even if the initial configuration does not translate to a connected topology. The final configuration figure also represents the effectiveness of our proposed collision avoidance method, as the agents have converged surrounding the rendezvous point.



**Figure 6.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with a single static maximum at the center (left) and their final configuration (right). Only one communications tower was used - marked by the blue square.

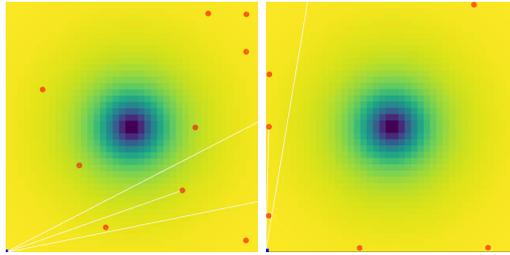
Figures 7(a) and 7(b) depict a mission plane with a single moving rendezvous target, type 1b, moving randomly from its initial position at the center of the mission plane. This simulation type is shown to represent that agents achieve their primary goal of rendezvousing, and their secondary of doing so in the highest-quality area, tracking it dynamically while it moves over the simulation.

The scenarios with single minimums - 2a and 2b - are represented in Figures 8(a) to 9(b). A direct consequence of having a single point with the lowest utility value

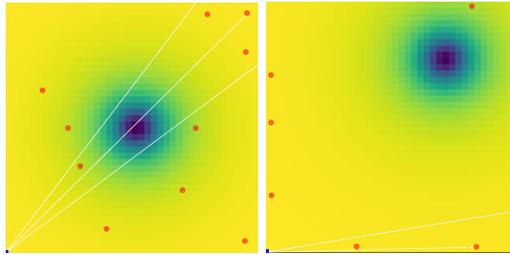


**Figure 7.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with a single moving maximum at the center (left) and their final configuration (right). Only one communications tower was used - marked by the blue square.

is the existence of multiple equally-desirable areas that span across most of the area of the mission plane. These simulations show that agents move away from the low-desirability area, despite it moving for Figures 9(a) and 9(b). In both cases, the final configuration is not constituted by a single cluster, as there is no single most-desirable area.



**Figure 8.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with a single static minimum at the center (left) and their final configuration (right). Only one communications tower was used - marked by the blue square.



**Figure 9.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with a single moving minimum at the center (left) and their final configuration (right). Only one communications tower was used - marked by the blue square.

More advanced scenarios are depicted in Figures 10(a) to 15(b), which contain multiple rendezvous areas (utility function maxima) and areas to be avoided (minima of the function) such as dynamic environmental obstacles.

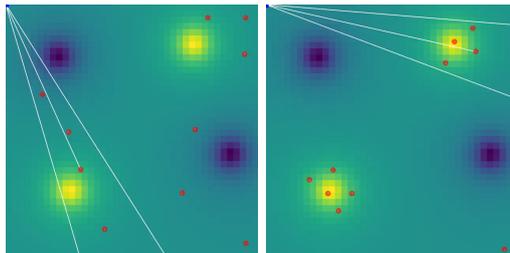
The simplest of these scenarios is illustrated in Figures 10(a) and 10(b), in which neither the rendezvous areas nor the minima move with time, which demonstrates the agents rendezvous to the multiple targets while avoiding the low-utility areas without colliding. Figures 11(a) and 11(b) represent an analogous scenario, with the addition of forbidden zones for the agents. Identically to the previous scenario, the

agents achieve their objectives, avoiding both the low-utility areas and the illegal zones. Moreover, these two scenarios are exemplified in environments with large networks (due to computational requirements the *Crazyswarm* simulations were capped at 180s and while the video shows the agents converging on the objective, the end of the simulation does not represent the final configuration. Hence we show the results for the *Matlab* simulations.) - 30 agents,  $200 \times 200$  mission plane, and 10 towers - in Figures 16(a), 16(b), and 17(a), 17(b), respectively.

Simulations of type 3e and 3f, shown in Figures 12(a) to 14(b), have the additional characteristic of time-varying utility values for the rendezvous areas to emulate surveillance missions where the utility decreases when there is an agent in its proximity and increases otherwise. If the value decreases below a constant threshold, the rendezvous target is removed from the mission plane and a new one is created at an arbitrary position. Figures 14(a) and 14(b), and 12(a) and 12(b) represent the cases with and without illegal zones in the mission plane, respectively. In both simulation examples the agents converge to the multiple rendezvous areas, while avoiding the low-utility zones and the illegal zones. In these specific cases, the agents are capable of exploring an area to the point its utility value is below the threshold, causing it to disappear, and proceed to a new one as a group. Figures 13(a) and 13(b) show scenario 3e - dynamic-value rendezvous points (without illegal zones) - with multiple communication towers. Due to the presence of an additional tower, compared to the simulation in Figures 12(a) and 12(b), the agents were able to converge to a single cluster.

In Figures 12(a) to 15(b), the agents final configurations appear to be in clusters outside of the rendezvous areas. We attribute this to the dynamism of their utility values: the final configurations represent the agents in an intermediary state between having explored a rendezvous area to the point it was removed and having reached a new one. This can be observed in the simulation videos.

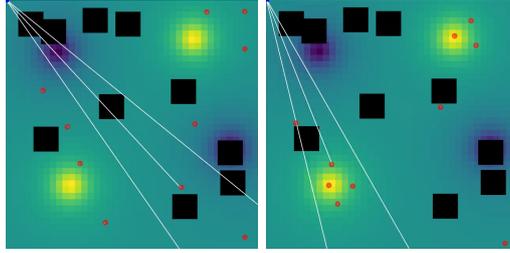
A more stochastic environment, type 3g, is presented in Figures 15(a) and 15(b), where the utility values of the rendezvous areas are dynamic and at each discrete time step, there is a probability that a rendezvous target (maximum point) or minimum target (minimum point) is created or removed. This simulation demonstrates that agents minimize the distance traveled to reach a new rendezvous area after the current one is removed.



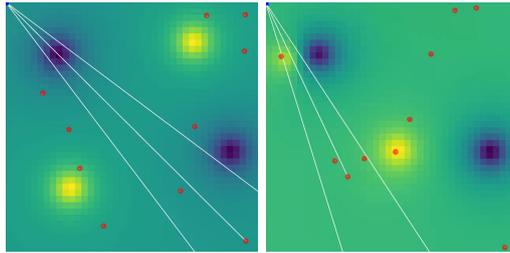
**Figure 10.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with multiple static rendezvous areas and multiple static minimums (of the utility function) (left) and their final configuration (right). Only one communications tower was used - marked by the blue square.

## 7. Conclusions and Future Work

This paper focused on the rendezvous problem for a group of mobile agents attempting to converge to multiple dynamic targets in a decentralized system. The agents were only assumed to be equipped with odometry sensors and antennae to receive the



**Figure 11.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with illegal zones (black squares) and an utility function with multiple static maximums (rendezvous areas) and multiple static minimums (left) and their final configuration (right). Only one communications tower was used - marked by the blue square.



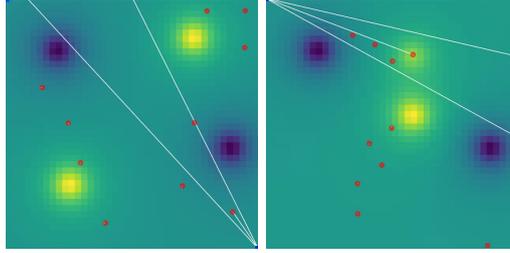
**Figure 12.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with multiple static rendezvous areas whose utility value decreases when an agent explores it, and multiple static minimums (of the utility function) (left) and their configuration at  $t=170s$  (right). Only one communications tower was used - marked by the blue square.

towers directional broadcasts containing state estimates. The proposed solution does not require synchronous communication nor the network topology to be connected. Instead of a consensus-based approach, we investigate the use of flocking-based movement rules, with added components tailored to our scenario. We also introduced a mechanism to avoid collisions between agents and proved convergence to a single cluster, whenever only one rendezvous target is available and the agent density allows it. The system considers imperfect measurements by working with set-valued estimates of the real positions, defined to be convex polytopes. It is shown through multiple simulations, which represent a multitude of scenarios with real-world applications, the effectiveness of our algorithm, and its limitations.

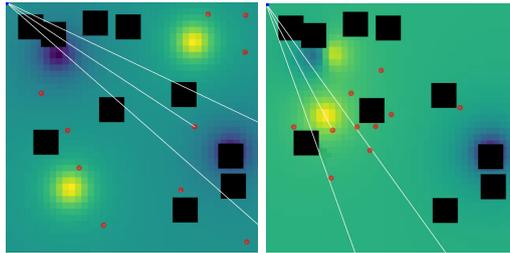
For future work directions, we consider:

- given that the SVOs are used for updating the set-valued estimates, one could propagate these sets using the same techniques and apply Model Predictive Control (MPC) strategies to better decide on the actuation;
- adding a scheduling to the communication towers to improve the convergence rate, opposed to round robin;
- implementing a formation component to be able to have structured formations and maximize the exploration area.

The first topic can use the propagations of the set-valued estimates for the neighbors and add a prediction horizon since the decision action is shared among the nodes. Moreover, the cost function for the MPC could take into account the average utility within the estimates. In doing so, each node would computed an optimized control action that drives towards maximizing the perceived utility within the possible future location and removing some of the oscillatory behavior of the trajectories.



**Figure 13.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with multiple static rendezvous areas whose utility value decreases when an agent explores it, and multiple static minimums (of the utility function) (left) and their final configuration (right). Two communications towers were used - marked by the blue squares.



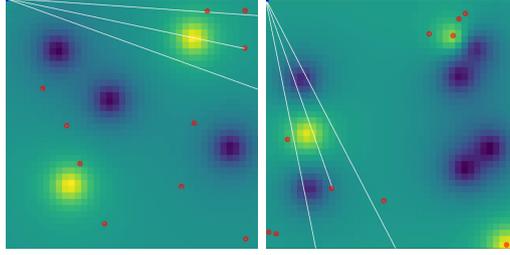
**Figure 14.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with illegal zones (black squares) and an utility function with multiple static maximums (rendezvous areas, whose utility value decreases when an agent explores it) and multiple static minimums (left) and their final configuration (right). Only one communications tower was used - marked by the blue square.

The current proposal uses round robin but this could be replaced by a method that finds a compromise between the estimation errors that each node will have for all its neighbors and also prioritize the vehicles with fewer neighbors in a given clique. The formations could also be achieved by having a common decision rule on how each node should compute its position within the formation and then add a flocking rule to move the vehicle towards the location.

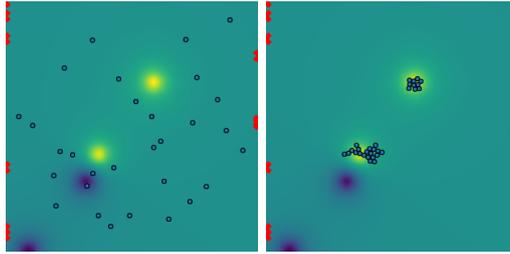
These future research directions could contribute to have a MAS that is capable of self-deploying in a region and take optimized actions to improve the future utility values regardless of the estimation errors associated with the noisy measurements. This future method would also be better at efficiently utilizing the communication protocol to update the estimates and drive nearby vehicles in a formation to improve the quality of readings of the utility function.

## References

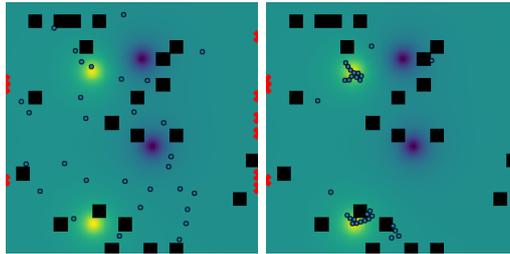
- Antunes, D., Silvestre, D., & Silvestre, C. (2011, Dec). Average consensus and gossip algorithms in networks with stochastic asymmetric communications. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)* (p. 2088-2093).
- Carli, R., Bullo, F., & Zampieri, S. (2010). Quantized average consensus via dynamic coding/decoding schemes. *International Journal of Robust and Nonlinear Control*, *20*(2), 156–175.
- Dimarogonas, D., & Johansson, K. (2009, December). Event-triggered control for multi-agent systems. In *48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference (CDC/CCC)* (p. 7131 -7136).



**Figure 15.** Initial MAS with  $n = 10$  agents in a  $20m \times 20m$  mission plane with multiple dynamic-value rendezvous zones and multiple static minimums (left) and their configuration at  $t=145s$  (right). At each discrete time step there is a probability that an utility function maximum or minimum is created or removed. Only one communications tower was used - marked by the blue square.



**Figure 16.** Initial MAS with  $n = 30$  agents in a  $200 \times 200$  mission plane with multiple static-value rendezvous zones and multiple static minimums (left) and their final configuration (right). Ten communication towers were used - marked by the red Xs. (*Matlab* simulation)



**Figure 17.** Initial MAS with  $n = 30$  agents in a  $200 \times 200$  mission plane with illegal zones (black squares) and an utility function with multiple static-value maximums - rendezvous zones - and minimums (left) and their final configuration (right). Ten communication towers were used - marked by the red Xs. (*Matlab* simulation)

Dimarogonas, D. V., Frazzoli, E., & Johansson, K. H. (2010, December). Distributed self-triggered control for multi-agent systems. In *49th IEEE Conference on Decision and Control (CDC)* (p. 6716 -6721).

Fagnani, F., & Zampieri, S. (2009). Average consensus with packet drop communication. *SIAM Journal on Control and Optimization*, *48*(1), 102-133.

Olfati-Saber, R., & Murray, R. (2004, September). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, *49*(9), 1520 - 1533.

Patterson, S., Bamieh, B., & El Abbadi, A. (2010, April). Convergence rates of distributed average consensus with stochastic link failures. *IEEE Transactions on Automatic Control*, *55*(4), 880 -892.

Reynolds, C. (1987, 07). Flocks, herds, and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, *21*, 25-34.

Ribeiro, R., Silvestre, D., & Silvestre, C. (2020, Sep). A rendezvous algorithm for multi-agent

- systems in disconnected network topologies. In *28th Mediterranean Conference on Control and Automation (MED)* (p. 592-597).
- Ribeiro, R., Silvestre, D., & Silvestre, C. (2021). Decentralized control for multi-agent missions based on flocking rules. In J. A. Gonçalves, M. Braz-César, & J. P. Coelho (Eds.), *Controlo 2020* (pp. 445–454). Cham: Springer International Publishing.
- Sadikhov, T., Haddad, W., Goebel, R., & Egerstedt, M. (2014, June). Set-valued protocols for almost consensus of multiagent systems with uncertain interagent communication. In *American Control Conference (ACC), 2014* (p. 4002-4007).
- Shamma, J., & Tu, K.-Y. (1999, feb). Set-valued observers and optimal disturbance rejection. *IEEE Transactions on Automatic Control*, *44*(2), 253 -264.
- 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.
- Silvestre, D. (2022b). Set-valued estimators for uncertain linear parameter-varying systems. *Systems & Control Letters*, *166*, 105311. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0167691122001141>
- Silvestre, D., Hespanha, J. P., & Silvestre, C. (2019, June). Broadcast and gossip stochastic average consensus algorithms in directed topologies. *IEEE Transactions on Control of Network Systems*, *6*(2), 474-486.
- Silvestre, D., Rosa, P., Hespanha, J. P., & Silvestre, C. (2015, July). Set-consensus using set-valued observers. In *American Control Conference (ACC), 2015, Chicago, Illinois, USA*.
- Silvestre, D., Rosa, P., Hespanha, J. P., & Silvestre, C. (2017). Stochastic and deterministic fault detection for randomized gossip algorithms. *Automatica*, *78*, 46 - 60.