Robotica (2026), **44**, pp. 1135–1158
doi:10.1017/S0263574726103506

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Chain-flocking: navigation of multiple agents with local measurements in unknown environments

Monica Rodriguez-Ortiz, Hector M. Becerra and Rafael Murrieta-Cid

Computer Science Department, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico
Corresponding author: Hector M. Becerra; Email: hector.becerra@cimat.mx

Received: 10 August 2025; **Revised:** 28 March 2026; **Accepted:** 28 April 2026; **First published online:** 1 June 2026

Keywords: consensus; collision avoidance; flocking navigation; multi-agent systems

Abstract

In this article, we propose a navigation approach based on a flocking algorithm for multiple agents that discover the environment with their sensors. The proposed method is able to move the agents in a flexible formation that preserves a pattern that minimizes a measure of energy, but is also able to avoid collision with obstacles while maintaining communication and visibility between agents. Control laws are combined with a new version of the planner called the Nearness Diagram (ND), extended to multiple agents. Two central ideas in the approach are: (1) the global leader of the formation can change, selecting as global leader one that has favorable conditions to reach the goal, (2) each agent locally follows another agent, the agent that is followed minimizes a cost related to the shortest path to the global leader in a graph. The method is intensively validated with several computer simulations, implemented in differential-drive robots. The results showed that the approach maintains communication formation and flexible enough to allow agents to line up when narrow passages appear in the environment.

1. Introduction

In recent years, multi-agent systems (MAS) have gained a notorious importance due to their ability to perform distributed tasks, such as exploration, surveillance, and transportation. The first issue that arises when working with MAS is the problem of moving a group of agents from one place to another without crashing, either with other agents or obstacles. Different formation types and algorithms have been proposed to tackle this issue [1, 2], but one that sticks out because of its amorphous properties is *flocking*. The flocking formation is a collective behavior where a large number of agents move together toward a common objective. This behavior has been noticed in nature and studied by diverse scientific disciplines like biology, physics, medicine, and even social science, due to the behavior that naturally emerges when large groups of individuals interact [3–5].

The main ideas of flocking were introduced by Reynolds in ref. [6], and then they used it in ref. [7] to obtain a stable control law. Later, Olfati-Saber [8] presented a complete theoretical framework for the design and analysis of the control law, and also included the possibility of controlling the flocking in a desired route while avoiding obstacles. The aim of this paper is to propose a solution for navigating a MAS through unknown GPS-denied and complex environments. We employ the flocking formation approach to achieve formation scalability and eliminate the need for pre-configurations. The control law for the flocking formation is built upon the algorithm in ref. [8], but avoiding the dependency of a global positioning system. Additionally, we introduce a cooperative implementation of a reactive obstacle avoidance method called Nearness Diagram (ND)[9, 10]. This method proposes a solution path to navigate in unknown environments while avoiding the visible local minima present in concave-shaped obstacles. Each agent suggests a potential path toward the goal; however, only the most suitable agent is designated as the formation leader, with the remaining agents assuming follower roles. Rather than strictly implementing a leader–follower formation, a shortest path algorithm is proposed, so each agent

© The Author(s), 2026. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

is able to determine its optimal neighbor to follow within the formation network directed toward the formation leader. This gives rise to a multi-leader–follower formation strategy named *chain-flocking*.

1.1. Related work

Numerous research works have addressed the various challenges inherent to MAS formations [11], such as network time delays, connectivity preservation, and obstacle avoidance, among others. In navigation of unknown environments, effectively solving the obstacle avoidance issue is crucial for task completion. In ref. [8], obstacle avoidance was accomplished by implementing a repulsive artificial potential function (APF) and the creation of a virtual agent to align with the obstacle's boundary. However, this is a decentralized method with the disadvantage of not considering other agents' obstacle information, potentially leading to issues like formation splitting or agents becoming trapped in local minima of the APF or concave obstacles. In an alternative approach [12], a stream function was introduced to avoid the APF local minima when the obstacle and goal path are aligned. Obstacle avoidance using consensus theory is achieved in ref. [13], where obstacle information is shared among agents to collectively avoid an obstacle without formation splitting. A similar behavior is presented in ref. [14] drawing inspiration from the starlings' behavior, where dynamic obstacles are also considered. In ref. [15], a bio-inspired algorithm for swarm path formation to explore an unknown environment using local sensor information is proposed. A fully distributed continuous region-reaching controller for multi-robot systems is proposed in ref. [16]; the robots can move cohesively, reaching velocity matching and collision avoidance within the moving region.

In ref. [17], the feasibility of achieving flocking behavior with only a subset of informed agents was demonstrated, leading to the proposal of a leader–follower formation scheme. Some works have proposed flocking algorithms without distinguishing between robots and obstacles [18, 19]. The obstacle avoidance problem has also been addressed as a connectivity issue. A control method was introduced in ref. [20] to address this connectivity challenge; network topology is adjusted when obstacles are encountered by deactivating links between agents, guided by geometric constraints between neighbors. Similarly, [21] focuses on identifying critical neighbors by analyzing agent's displacements, preventing disconnections that could trigger a global network disruption.

1.2. Contributions

The contributions of our work can be enumerated as follows:

1. A local leader–follower second-order control law is proposed to avoid the use of a global positioning system, allowing the formation to operate in GPS-denied environments. In contrast, most of the related works, such as [8, 12, 13, 16, 17], use a positioning system. This is achieved by using a multi-leader–follower scheme named *chain-flocking*, where every agent searches for its own local leader between its neighbors by using the shortest path toward the formation leader inside the MAS network. Although existing approaches [18, 19], which do not distinguish between robots and obstacles, rely on local measurements, they are limited to dealing with convex or nearly convex obstacles.
2. All agents are doted with the capability of calculating a local route toward the goal using an ND [9] *local motion planner* extended to multiple agents for 2D scenarios. This avoids the dependency on a predefined leader as proposed in refs. [8, 17, 19] or a predefined navigation route as formulated in refs. [18, 20, 21]. In our approach, to select the best-suited agent toward the goal, the agents deliberate with the path information received by the neighbors to select the global formation leader. The motion planning strategy is a key feature of our approach that is not present in any of the related works. For 3D scenarios, other planners capable of handling 3D obstacles can be used; thus, the proposed method is not limited to a particular planner.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

3. The agents are capable of avoiding visible local minima, due to the local path planning, but also due to the multi-leader–follower scheme, where the agents find their own obstacle-free path by following each other (chain-flocking), which is a feature of our method that does not exist in other approaches. Therefore, unlike the few works dealing with obstacles with general shapes, with our approach, a MAS is capable of navigating complex environments with non-convex obstacles not stucking in local minima, without using global information of obstacles [22] nor following the obstacles boundary [19].

In summary, the overall contribution of this paper is the proposal of control laws and algorithms that allow a MAS to navigate in a group from one initial position to a goal position in unknown, complex, and GPS-denied environments.

2. Preliminaries and problem formulation

This section provides some basics on the flocking framework, defines the motion model of the agents and their sensing capabilities, and finally states the addressed problem.

2.1. Flocking basics

The theoretical framework proposed by Olfati-Saber [8] defines agents (real and virtual) to induce different behaviors in the control law:

- α-agents are real neighbor agents that induce forces to maintain the formation.
- β-agents are virtual agents that enable obstacle avoidance.
- γ-agents are virtual agents that allow the formation navigation.

In that work, the agents are modeled by second-order particle dynamics. Each agent i is ruled by a control law composed of three terms:

$$u_i = u_i^{\alpha} + u_i^{\beta} + u_i^{\gamma} \tag{1}$$

Each term aims to induce behaviors for the corresponding kind of agent: u_i^{α} is a gradient-based term to induce separation, cohesion, and alignment between α-agents, u_i^{β} is a repulsion term in charge of avoiding obstacles, and u_i^{γ} is dedicated to tracking a moving point as navigational feedback. While α and β-agents are proximity agents depending on relative distances that can be represented in local reference frames, the virtual γ-agent represents a globally defined moving rendezvous point for all the formation. The application of these navigation rules is not possible under a system with no global position reference.

2.2. Agents model

We will be considering n number of homogeneous agents with second-order particle dynamics:

$$\dot{q}_i = p_i, \quad \dot{p}_i = u_i \tag{2}$$

where $q_i, p_i, u_i \in \mathbb{R}^m$, are the position, velocity, and control input vectors, respectively. Each agent is defined by a unique ID and should be able to calculate its own velocity. All agents share a common global direction but not a common global positioning system; therefore, each agent uses a local reference frame oriented in a global direction.

2.3. Sensing capabilities

We assume that all agents can communicate among them through a communication network, modeled as a graph $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}(t)\}$ where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of vertices that represent each agent and $\mathcal{E}(t) \subseteq \{(i, j) \mid i, j \in \mathcal{V}, j \neq i\}$ is the set of edges, which represent the relations between these agents.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

Besides, each agent carries an omnidirectional range sensor and a compass. The agents move in an environment populated with obstacles, which will be denoted by the set $\mathcal{O}$. The agents do not know a priori the location of the obstacles; they discover the obstacles with their sensors. The visibility capability allows the agent to visualize other agents and obstacles and calculate its relative measurements.

Assumption 1. *All agents have identical perception capabilities with the same maximum distance range r_α^{max}; therefore, they perceive each other at the same time.*

Assumption 2. *All agents are located in the free space region q_i at the start time.*

Each agent i will generate a visibility polygon $\mathcal{P}_i$ from its position q_i, and it is defined as the set of points q that are visible from point q_i, that is, the line segment joining q and q_i that has no intersection with any point of any obstacle:

$$\mathcal{P}_i = \{q \in \mathbb{R}^m \mid \overrightarrow{qq}_i \cap \mathcal{O} = \emptyset\} \tag{3}$$

Definition 1. *The perception zone of agent i is denoted as $\mathcal{P}_i^\alpha$, and is defined as the visibility polygon constrained by the maximum perception range,*

$$\mathcal{P}_i^\alpha = \{q \in \mathcal{P}_i \mid q < r_\alpha^{max}\} \tag{4}$$

Definition 2. *Every agent i can detect the other agents that lie inside the perception zone $\mathcal{P}_i$ and form the $\mathcal{N}_i$ neighborhood. This set will be called perceived neighbors,*

$$\mathcal{N}_i^\alpha = \{j \in \mathcal{V} \mid q_j \in \mathcal{P}_i^\alpha\} \tag{5}$$

An agent cannot detect another agent that is blocked by an obstacle.

Definition 3. *Problem statement. The objective of this work is to design a flocking protocol that allows to accomplish the following three tasks for a group of agents modeled as in (2):*

1) *achieve identical inter-agent distance r_α^{des} over a network $\mathcal{G}(t)$, that is, $\|q_i - q_j\| = r_\alpha^{des}, \forall j \in \mathcal{N}_i^\alpha$, which means to form an α-lattice geometric pattern;*
2) *avoid fixed obstacles in the environment, maintaining a safety distance r_β^{des}, that is, $\|q_i - q_{o_i}\| < r_\beta^{des}, \forall q_{o_i} \in \mathcal{O}$;*
3) *move toward a navigational direction p_d defined by a selected leader, that is, $\|p_i - p_d\| \to 0$ as $t \to \infty$.*

These tasks must be accomplished by using measurements only from onboard sensors.

3. Proposed solution

In this section, we define desired agent behaviors, the control laws to achieve those behaviors, and a motion planning method, such that navigation in flocking can be realized in unknown environments.

3.1. Agent behaviors

First, we define similar types of agents as in the flocking basics, α-agents, β-agents, and γ-agents, but with important differences with respect to existing approaches [8, 17–19].

3.1.1. Modeling connectivity: α-agents

For flocking, the agents need to form a connected graph to achieve cohesion and to build and α-lattice structure. Typically, existing works consider that two agents are connected if they are inside the maximum distance range r_α^{max}. Moreover, the perception range is restricted to accomplish $\frac{r_\alpha^{max}}{r_\alpha^{des}} = 1 + \epsilon$ with $\epsilon \ll 1$. We propose to enlarge the perception range without affecting α-lattice feasibility by using a

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

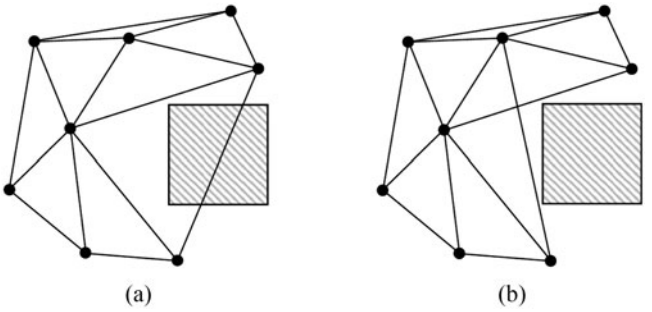


Figure 1. *Agents defining Delaunay graphs. (a) Delaunay graph $\mathcal{G}_D$ without restrictions due to obstacles and maximum distance. (b) α-Delaunay graph built with constraints imposed by obstacles and maximum range.*

Delaunay triangulation [23] to define the network topology. In a proximity graph, an agent's neighborhood is defined by $\mathcal{N}_i(t) = \{j \in \mathcal{V} \mid r_{ij} < r_\alpha^{max}\}$, where r_{ij} is the distance between agents i and j. For a large number of agents and a large perception range, the proximity graph leads to communication overload and geometric inconsistencies in forming the α-lattice structure. Agents that are distant and obstructed by other agents will never be able to attain the desired distance value.

A Delaunay graph $\mathcal{G}_D(t) = (\mathcal{V}, \mathcal{E}_D(t))$ is a geometric structure that connects a set of points in space with nonoverlapping edges [23]. The basic properties that make a Delaunay graph a better choice for connectivity definition in comparison with a proximity graph are: (1) The existence of a spanning tree in the network is guaranteed. (2) There is only one Delaunay graph for a given set of points, and it can be calculated locally. (3) The empty circumcircle property, defined as any three vertices connected by edges that do not contain any other vertex inside, allows the construction of the connectivity graph. Figure 1 shows agents connected in Delaunay graphs without restrictions and with constraints imposed by obstacles and maximum distance.

3.1.2. Obstacles detection: β-agents

The agents are not informed about the set of obstacles $\mathcal{O}$ in the environment; instead, they will identify a portion of the set when an obstacle is within their visibility range r_α^{max}. The obstacles and agents will induce different behaviors; therefore, it becomes necessary to distinguish between obstacles and other agents within the environment.

Assumption 3. *The agents are able to differentiate between obstacles and other agents.*

This distinction is possible due to the information exchanged among agents through the communication network (all can communicate with each other). Each agent will explore the environment, seeking obstacles in the environment and generating a subset $\mathcal{O}_i$ based on the maximum obstacle perception range $r_\beta^{max} \leq r_\alpha^{max}$. This subset may contain multiple obstacles, which are distinguishable due to distance discontinuities.

Each agent with position q_i defines its own obstacles set $\mathcal{O}_i$ as follows:

$$\mathcal{O}_i = \{q_o \in \mathcal{O} \mid \|q_i - q_o\| < r_\beta^{max}\} \tag{6}$$

and can differentiate between κ distinct obstacles $\mathcal{O}_{ik}$, $k = \{1, \ldots, \kappa\}$, by detecting a distance discontinuity η:

$$\mathcal{O}_{ik} = \{q_l \in \mathcal{O}_i | \forall q_l \in \mathcal{O}_{ik}, \exists q_s \in \mathcal{O}_{ik}, l \neq s, \|q_l - q_s\| \leq \eta\} \tag{7}$$

where $q_l, q_s \in \mathcal{O}_i$ are two points from the obstacles perceived set, and obstacle point q_l is considered to be part of the obstacle $\mathcal{O}_{ik}$ if there exists another point q_s in that obstacle such that the distance between them is less than or equal to the distance discontinuity η.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

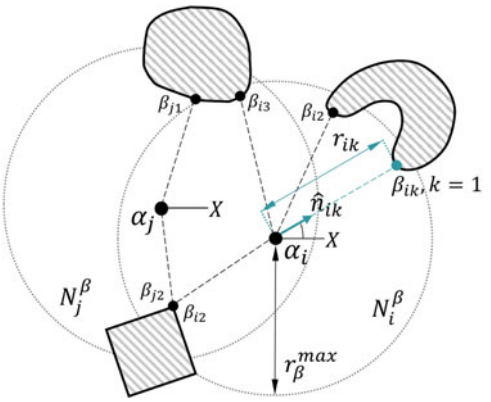


Figure 2. *Obstacles detection and definition of β-agents.*

Each obstacle $\mathcal{O}_{ik}$ will be represented by the closest point of the obstacle to agent i, and this point will form a virtual agent at the boundary of the obstacle, which will be called β-agent. See Figure 2.

Definition 4 (β-neighborhood). *The set $\mathcal{N}_i^\beta$ of β-agents of agent i is created at the closest point between each obstacle $\mathcal{O}_k \in \mathcal{O}_i$ and agent i:*

$$\mathcal{N}_i^\beta = \{q_o \in \mathcal{O}_{ik} \mid q_o = \arg\min_{q \in \mathcal{O}_{ik}} (\|q_i - q\|)\} \tag{8}$$

The obstacle point q_o is employed only for definition purposes; however, it is possible to directly acquire relative information about the obstacle, such as the relative distance r_{ik} and angle θ_{ik} of the closest point q_o. The unit vector from the agent to the closest point is given by $\hat{n}_{ik} = [\cos\theta_{ik}, \sin\theta_{ik}]^T$. This relative information will be used by the control law for obstacle avoidance.

3.1.3. Chain-flocking: γ-agents

In this work, due to the absence of a global positioning system, unlike existing works [8, 18], we rely on the local perception of each agent and the information that can be exchanged within the network to define a navigation rule for every agent. We propose that agents follow each other, selecting a neighbor to track that leads them to a leader agent. This strategy is our main proposal and we call it *chain-flocking*.

The chain-flocking method is based on the visibility properties from the agent's perception. Now, assume that a formation leader γ_0 has already been selected (the selection process will be described in Section 3.1.4). If all agents are connected by the α-Delaunay graph $\mathcal{G}_{\alpha D}$, which posses the visibility constrains of Def. 1 (see Figure 3a), then each agent $i \in \mathcal{V}$ can find a graph path within $\mathcal{G}_{\alpha D}$ that is free of obstacles connecting with the leader γ_0. This is illustrated in Figure 3b, where it is visible that each connected agent can keep on track of the leader by tracking its own neighbor. Consequently, if the leader is informed about a path that is free of obstacles, then each follower agent is able to navigate the same path as the leader by indirectly tracking it through a neighbor that connects with it. This neighbor will be called γ-agent, and each agent i identifies its own γ_i-agent to follow. To do so, two criteria must be considered: (1) avoid risk of collision with any other agents during the tracking, and 2) select the nearest neighbor to the leader within the formation graph.

To address the first criteria, the *Gabriel graph* proximity notion is introduced. Risk of collision is avoided by introducing a constraint in the agents' connectivity to form a new graph called the Gabriel graph, formed theoretically as the intersection of the edges of the Delaunay graph and the Voronoi diagram. In practice, the condition for connecting two agents in the Gabriel graph is to establish a *region of influence* where no other agent should reside, as illustrated in Figure 4. This region is defined by a circle with center at the midpoint of the two agents and circumscribed by them. The neighbors that fulfill the Gabriel condition will be called Gabriel neighbors $\mathcal{N}_i^{\alpha G}$.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

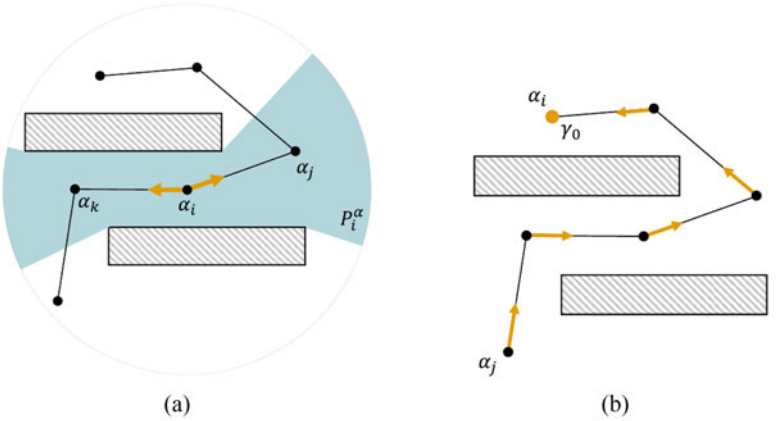


Figure 3. *Visibility and chain path for navigation of the MAS. (a) Perception zone. (b) Chain path toward the leader.*

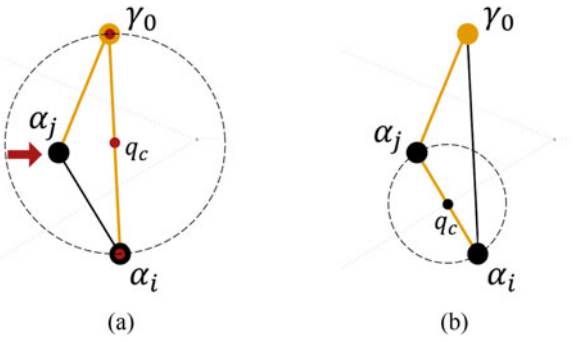


Figure 4. *Gabriel condition of Def. 5. (a) Condition not fulfilled: agent $\gamma_0 \notin \mathcal{N}_i^{\alpha G}$, as the region of influence between γ_0 and α_i is occupied by α_j. (b) Condition fulfilled: agent $\alpha_j \in \mathcal{N}_i^{\alpha G}$, as the region of influence between α_i and α_j is free.*

Definition 5 (Gabriel neighborhood [24]). *Given a neighbor $j \in \mathcal{N}_i^{\alpha D}$ of agent i, j will be part of the Gabriel neighborhood $\mathcal{N}_i^{\alpha G}$ if there exists a circle c with center at the midpoint $h = \frac{q_i - q_j}{2}$ and circumscribed by agents i and j, such that there is no other agent $l \in \mathcal{N}_i^{\alpha D}$ positioned inside the circle c, that is:*

$$\mathcal{N}_i^{\alpha G} = \{j \in \mathcal{N}_i^{\alpha D} | \, \forall l \neq j \in \mathcal{N}_i^{\alpha D}, \|h - q_j\| \leq \|h - q_l\|\} \tag{9}$$

This set will be used by each agent i to find its γ-agent to follow. Once the collision avoidance criterion has been established, the agent to follow γ_i must comply with the nearest neighbor criterion. The nearest neighbor is defined as the neighbor with the minimum distance toward the leader agent γ_0 within the formation graph $\mathcal{N}_i^{\alpha G}$. This criterion is evaluated by using the distances between agents as the weights of the graph edges to allow the search for the shortest path in distance inside the graph. As every agent knows only local information about their neighbors, the shortest path length is computed in a distributed manner using distances shared through the communication network.

The shortest path length $r^*_{i\gamma_0}$ of agent i is calculated by expanding the shortest path length $r^*_{j\gamma_0}$ of a neighbor j by adding the sensing distance to the neighbor r_{ij}, and finding the minimum among all expanded paths through the graph. See example in Figure 5a.

By using the shortest path length over the Gabriel Graph neighbors $j \in \mathcal{N}_i^{\alpha G}$, we address both restrictions for the selection of the γ_i-agent: avoiding collisions with neighbors and ensuring proximity to the

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

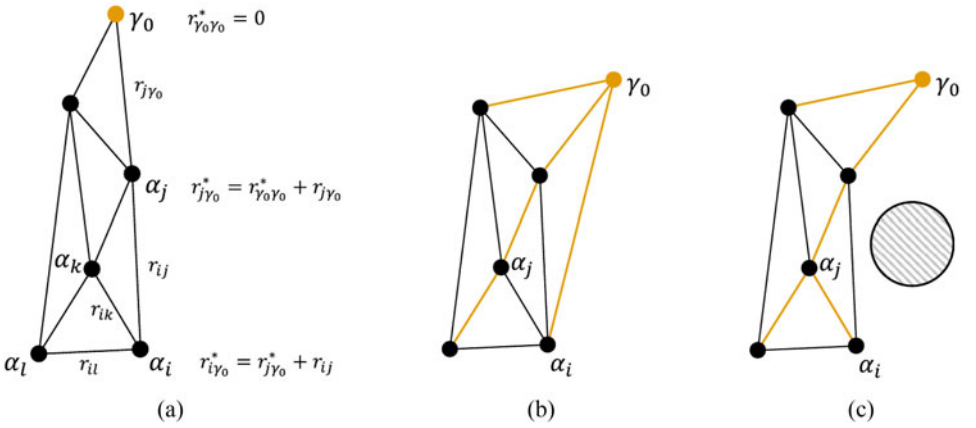


Figure 5. *Shortest path length. (a) Example of computation. (b) $\gamma_i = \gamma_0$, as there are no obstacles. (c) $\gamma_i = \alpha_j$, as there is one obstacle.*

leader. However, the collision avoidance restrictions are relaxed when there are no obstacles sensed by the agent $\mathcal{N}_i^{\beta} \neq \emptyset$, as even if the agent passes near another agent, it has free space to avoid collision, which is not the case when there are obstacles near the agent. This difference is illustrated in Figure 5b, c. Then, each agent can obtain its α-agent by applying the following definition:

Definition 6 (γ-agent). *The γ-agent of agent i is its neighbor j that belongs to Gabriel neighborhood $j \in \mathcal{N}_i^{\alpha G}$ when there are sensed obstacles $\mathcal{N}_i^{\beta} \neq \emptyset$, or to the α-Delaunay neighborhood $j \in \mathcal{N}_i^{\alpha D}$ otherwise, and that has the shortest path length to the leader agent γ_0. This is described as:*

$$\gamma_i = \begin{cases} \arg\min_{j \in \mathcal{N}_i^{\alpha G}} \left(r^*_{j\gamma_0} + r_{ij} \right), \\ \arg\min_{j \in \mathcal{N}_i^{\alpha D}} \left(r^*_{j\gamma_0} + r_{ij} \right), \end{cases} \tag{10}$$

where $r^*_{j\gamma_0}$ is the shortest path length of neighbor j to the leader γ_0, which is received through communication, and r_{ij} is the distance to the neighbor j sensed by the agent i.

Now, we can define the formation graph given by the γ-agents by considering all the relations between each agent and its γ_i-agent.

Definition 7 (Chain graph). *The chain graph $\mathcal{G}_C(\mathcal{V}, \mathcal{E}_C)$ is a directed spanning tree with root node at the leader agent γ_0, which allows the distribution of the information from the leader to every agent $i \in \mathcal{V}$. The edges of this graph are defined as $\mathcal{E}_C = \{(i, \gamma_i), \forall i \in \mathcal{V}\}$, that is, the edges that connect every agent i with its corresponding γ-agent. For instance, see the yellow edges in Figure 5b, c.*

3.1.4. Navigation γ_0-agent (leader)

To avoid the dependence of a predefined leader–follower formation, we propose to select as leader the best-suited agent toward the goal. This can be achieved assuming that all agents must be informed about the desired goal. Besides, the agents must be capable of sharing their information among their neighbors, so they can compare with each other and determine the best-positioned agent relative to the goal.

We propose a method for selecting a formation leader in a distributed manner. A cost function J^g is used to identify the agent with the minimum cost toward the goal. The selected leader will have a motion planning strategy to navigate toward the goal. The definition of the cost to goal criteria J^g will depend on information data from the environment, such as estimated path error to goal, passage sizes, obstacle sizes, etc. The cost function J^g and the motion planning strategy will be described later in Section 3.3.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

Algorithm 1. Leader selection

```
for each agent, i ∈ V do
    Calculate a path using the motion planning method (Section 3.3).
    Calculate cost to goal J_i^g according to motion planning.
    Receive shared costs from neighbors {J_j^g, j ∈ N_i^αD}.
    Find the agent with the minimum cost k* = arg min_k (J_k^g): k = {j ∪ i}.
    Propagate the minimum cost to goal J_i^g* = J_k*^g.
    if k* = i then
        Establish agent as leader γ_0 = i.
    else
        Establish agent as follower γ_0 ≠ i.
    end if
end for
```

Definition 8 (Leader γ_0). *The formation leader is the agent $i \in \mathcal{V}$ whose cost function J_i^g is the minimum of all agents, that is:*

$$\gamma_0 = i = \arg\min_{i \in \mathcal{V}} (J_i^g) \tag{11}$$

The agents cannot evaluate the cost function directly, as they only communicate with their neighbors. To select the leader, each agent i compares the cost shared by their neighbors J_i^g to find the agent with the minimum cost J_i^{g*} among them all, including itself. If the agent i founds itself as the minimum cost agent, then it becomes the formation leader and the propagated cost will be equal to its own cost $J_i^{g*} = J_i^g$; otherwise, the agent will propagate to its neighbors j the cost of a neighbor, or even the cost of an agent that is not within the perception range of agent i, whose cost is the minimum. This enables the agents to propagate the cost of the best-suited neighbor throughout the entire formation. This formulation is detailed in Alg. 1.

3.2. Control algorithm

Every agent $i \in \mathcal{V}$ applies a control input with the three components of Eq. 1, where each term determines the particular behavior induced by each agent type: α-control for formation, β-control for obstacle avoidance, and γ-control for navigation. These control laws are inspired by the work in ref. [8], but an important difference is that each control term uses local measurements obtained from onboard distance sensors. Moreover, the behavior of the flocking navigation is really different from the one in ref. [8], as was described in Section 3.1.3.

3.2.1. α-control (formation)

The purpose of the control term u_i^α is to drive the agents to form an α-lattice structure. The α-control is composed of two terms; the first one is a gradient-based term, and the second one is a consensus-based term:

$$u_i^\alpha = \frac{1}{|\mathcal{N}_i^{\alpha D}|} \left\{ \kappa_g^\alpha \sum_{j \in \mathcal{N}_i^{\alpha D}} \phi_\alpha(d_{ij}) \hat{n}_{ij} + \kappa_p^\alpha \sum_{j \in \mathcal{N}_i^{\alpha D}} a(d_{ij})(p_j - p_i) \right\} \tag{12}$$

where $\kappa_g^\alpha, \kappa_p^\alpha > 0$ are control gains, gradient and proportional, respectively, $d_{ij} = \|r_{ij}\|_\sigma$ is the distance between agents α_i and α_j under a σ-norm (see [8]), and $\hat{n}_{ij}$ is the unit vector connecting those agents. $\phi_\alpha(z)$ and $a(z)$ are adequate functions to define the potentials as in ref. [8]. The action function $\phi_\alpha(z)$ will drive the agents to a desired inter-distance r_α^{des}. To avoid inter-agent collision for agents with a body, the body size radius r_α^{size} needs to be considered in the selection of the desired distance between agents such that $r_\alpha^{size} < r_\alpha^{des} < r_\alpha^{max}$.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

3.2.2. β-control (obstacle avoidance)

The control term u_i^{β} allows the agents to avoid the obstacles in the environment. It creates a repulsive artificial potential field at the closest point of each perceived obstacle. The β-control is only composed of a gradient-based term:

$$u_i^{\beta} = \kappa_g^{\beta} \frac{1}{|\mathcal{N}_i^{\beta}|} \sum_{k \in \mathcal{N}_i^{\beta}} \phi_{\beta}(d_{ik}) \hat{n}_{ik} \tag{13}$$

where $\kappa_g^{\beta} > 0$ is a control gain for the gradient, $d_{ik} = \|r_{ik}\|_{\sigma}$ is the distance between agents α_i and β_{ik} under a σ-norm, and $\hat{n}_{ik}$ is the unit vector connecting those agents. $\phi_{\beta}(z)$ is an adequate function to define the potentials as in ref. [8]. The action function $\phi_{\beta}(z)$ will drive the agents to a desired distance from obstacles r_{β}^{des}, generating a repulsive force when the distance falls below the designated threshold. To avoid collision with obstacles for agents with a body, the body size radius r_{α}^{size} needs to be considered in the selection of the security distance between agents and obstacles such that $r_{\alpha}^{size} < r_{\beta}^{des} < r_{\beta}^{max}$.

3.2.3. γ-control (navigation)

The control term u_i^{γ} is different for the leader agent $i = \gamma_0$ and followers $i \neq \gamma_0$, each case is defined next.

–*Followers control*: The purpose of the control term u_i^{γ} is to follow the γ-agent selected by the method of Section 3.1.3. This term will lead indirectly the agent i toward the leader γ_0. The control law will maintain the neighbor agent i at a desired distance $r_{\gamma}^{des} = r_{\alpha}^{des}$ while tracking its velocity $p_{i\gamma}$. This control law drives the agents to form the chain-flocking structure. The γ-control is composed of two terms, a gradient-based term and a velocity consensus term:

$$u_i^{\gamma} = \kappa_g^{\gamma} \phi_{\gamma}(d_{i\gamma}) \hat{n}_{i\gamma} + \kappa_p^{\gamma} c(d_{i\gamma}) \left(p'_{i\gamma} - p_i\right), \quad i \neq \gamma_0 \tag{14}$$

where $\kappa_g^{\gamma}, \kappa_p^{\gamma} > 0$ are control gains, gradient and proportional, respectively, $d_{i\gamma} = \|r_{i\gamma}\|_{\sigma}$ is the distance between agents α_i and γ_i under a σ-norm, $\hat{n}_{i\gamma}$ is the unit vector connecting those agents and $p'_{i\gamma}$ is a target velocity vector that will be defined later. $\phi_{\gamma}(z)$ and $c(z)$ are adequate functions to define the potentials as in ref. [8].

The main goal of the control u_i^{γ} is to drive each follower toward its corresponding γ_i-agent with velocity vector $p_{i\gamma}$ to achieve the chain-flocking behavior. However, to deal also with obstacles, the velocity $p_{i\gamma}$ must be deviated by the obstacles perceived $\mathcal{N}_i^{\beta}$, resulting in the following target velocity:

$$p'_{i\gamma} = \begin{cases} \frac{1}{|\mathcal{N}_i^{\beta}|} \sum_{k \in \mathcal{N}_i^{\beta}} \left\{ b(\bar{\theta}_{ik}, d_{ik}) p_{ik} + (1 - b(\bar{\theta}_{ik}, d_{ik})) p_{i\gamma} \right\}, & \mathcal{N}_i^{\beta} \neq \emptyset \\ p_{i\gamma}, & \mathcal{N}_i^{\beta} = \emptyset \end{cases} \tag{15}$$

where $b(\bar{\theta}_{ik}, d_{ik}) \in [0, 1]$ is a function that allows the smooth transition between the γ_i agent's velocity $p_{i\gamma}$ and the virtual obstacle velocity p_{ik} and is defined as $b(\bar{\theta}_{ik}, d_{ik}) = \rho_{h_{\theta}}\left(\frac{\bar{\theta}_{ik}}{(2/3)\pi}\right) \rho_{h_{\beta}}\left(\frac{d_{ik}}{d_{\beta}^{max}}\right)$, where ρ_h are bump functions as in ref. [8], with $d_{ik} = \|d_{ik}\|_{\sigma}$ the distance between agent i and its β-agent β_{ik}, $d_{\beta}^{max} = \|r_{\beta}^{max}\|_{\sigma}$ the maximum obstacle detection range, both under the σ-norm. The parameter $\bar{\theta}_{ik}$ is the angle between the agent's direction $\hat{n}_i$ and the obstacle's direction $\hat{n}_{ik}$, which is given by $\bar{\theta}_{ik} = \cos^{-1}(\hat{n}_i \cdot \hat{n}_{ik})$. The function $b(\bar{\theta}_{ik}, d_{ik})$ allows an agent to overlook an obstacle when it is too far or when the agent is not navigating toward it.

In addition, each agent β_{ik} will create a virtual velocity p_{ik} to deviate the velocity $p_{i_{\gamma}}$ with the same magnitude, but perpendicular to the obstacle agent's direction $\hat{n}_{ik}$, that is:

$$p_{ik} = s_{ik} \hat{n}_{ik}^{\perp} \|p_{i_{\gamma}}\| \tag{16}$$

where s_{ik} allows the correct selection of the direction of the target velocity; $s_{ik} = 1$ if $\cos^{-1}\left(\hat{n}_{ik}^{\perp} \cdot \hat{n}_{i_{\gamma}}\right) \leq \frac{\pi}{2}$, and $s_{ik} = -1$ otherwise, with $\hat{n}_{ik}^{\perp}$ the vector perpendicular to the obstacle and $\hat{n}_{i_{\gamma}}$ the vector pointing to the γ_i agent.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

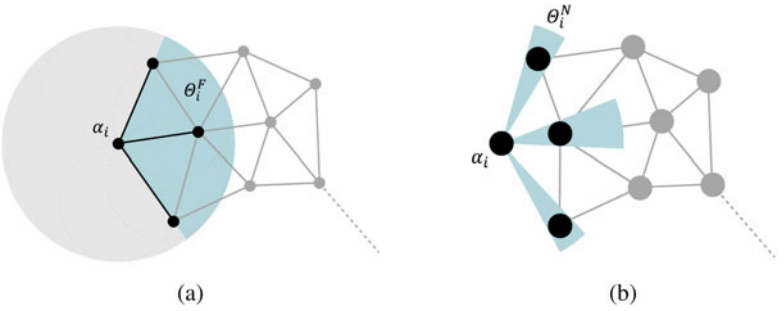


Figure 6. *Formation and neighbor's regions. (a) Formation region Θ_i^F of agent i. (b) Neighbor's regions Θ_i^N of agent i.*

–Leader control: We define a control law that requires only a velocity vector to guide the leader agent through the environment, not depending on a global positioning system. The γ-control of the leader is composed of one term that is a velocity tracking function defined as:

$$u_i^{\gamma} = \kappa_p^{\gamma_0}(p_i^g - p_i), \quad i = \gamma_0 \tag{17}$$

where $\kappa_p^{\gamma_0} > 0$ is a control gains. The velocity target p_i^g is calculated by the motion planning scheme, which is described in the following section.

Notice that an important difference between Olfati-Saber's method [8] and the proposed method is that u_i^{γ} are defined differently. In our method, the leader's motions deform the lattice, since each agent has a local leader, who is following, defined by a planning method. When the global leader reaches the goal, it is motionless, and eventually, each local leader will be motionless. At that time, the approach becomes exactly the same as the Olfati-Saber's one. Hence, the convergence of the formation is guaranteed when the goal is reached, similarly to the Olfati-Saber's method.

3.3. Motion planning

To allow the navigation of the agents without a global positioning system, we proposed the use of a reactive navigation method, the so-called ND [9]. We underline that it is possible to change only the motion planning method, adapted to our multi-agent scenario, while keeping the same control scheme. The main goal of the motion planning method is to obtain the target velocity vector for the leader and the metrics to select the leader using Alg. 1. The ND is a reactive obstacle avoidance method for 2D space, proposed in ref. [9].

In this work, we have extended the ND method to multiple agents. We use the so-called α-shape structure [25] to find the frontier of the formation under range constraints ($\alpha = r_{\alpha}^{max}/2$) since agents in the frontier have clear visibility of the environment. The α-shape is a generalization of the convex hull, described as the smallest geometric structure that encloses a set of points in space, given a maximum distance between agents.

Using the α-shape, *formation regions* are calculated. A formation region is composed of intervals of angles indicating the inside of the formation, which are clearly not traversable sectors. An example of a formation region is illustrated in Figure 6a. For agents with a body, it is necessary to define *neighbor regions* that consider the agents' size. An example of a neighbor region is shown in Figure 6b where it is visible that the region is expanded as the agent's size increases.

To avoid collision with moving agents and using the formation and neighbor regions, we propose the Formation Nearness Diagram (FND) from the central point, which is an extension of the PND,[1] as defined in ref. [9]. To detect obstacles in the FND, depth discontinuities (called gaps) are found. In the FND, when moving agents are close, blocking a path, they are merged together, creating a virtual

[1] Nearness Diagram from the central point (PND) establishes the distance from the center of the robot to each obstacle.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

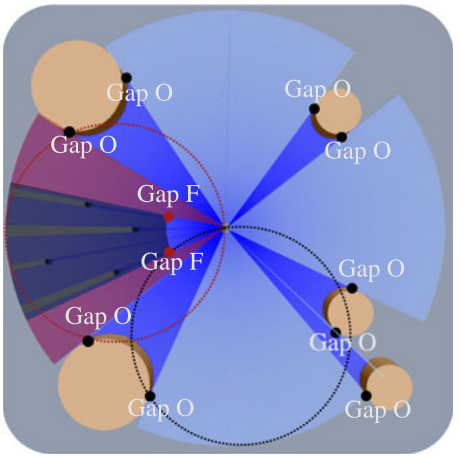


Figure 7. *Formation Nearness Diagram (FND), traversable and no-traversable valleys. Valleys in red are not traversable.*

obstacle. See Gap F in Figure 7. As in the original ND, valleys are formed by two discontinuities or gaps created by the obstacles in the environment. But in this work, to deal with moving agents that can block a path, a valley is not traversable if a circle formed between the agent and an obstacle's gap is not empty. See Figure 7. By applying the criteria on traversability defined in ref. [9] to our FND, the result is a navigation sector s_θ with direction angle φ_θ. This sector is collision-free and is related to the traversable valley that is closest to the goal direction.

The ability to change leaders ensures that at least one robot will always find a collision-free path to the goal if the path exists; then, as the robots move following the leader, they find a traversable valley.

We left for future work to extend the planner to the 3D case. However, an advantage of our method is that other planners capable of handling 3D obstacles can be used, as we will show in the implementation results. In other words, the proposed method is not limited to a particular planner.

3.3.1. Cost function for leader selection

As described in Section 3.1.4 and in Alg. 1, now we can define a cost function considering the features of the proposed planning method. Eq. 18 establishes the proposed cost function.

$$J_i^g(\theta_i^g, r_i^g, ID_i) = w_\theta|\varphi_\theta - \theta_i| + w_{\theta_g}|\varphi_\theta - \theta_i^g| + w_r\left(r_i^g - \min_{i\in V} r_i^g\right) + P + w_{ID}ID_i \tag{18}$$

where $w_\theta, w_{\theta_g}, w_r, w_{ID} \in \mathbb{R}$ are positive constants that define the weight for each term. The selection of the weights affects the following priorities:

1. w_θ minimizes path error to the current direction $|\varphi_i - \theta_i|$; preferring agents that calculate a navigation path that does not deviate the agent from its current direction. φ_θ is the navigation direction given by the motion planner, and θ_i is the i-th agent's heading in the global frame given by the onboard compass.
2. w_{θ_g} minimizes path error to the goal $|\varphi_\theta - \theta_i^g|$; preferring agents that calculate a navigation direction with less error angle toward the goal. θ_i^g is the heading of agent i with respect to the goal.
3. w_r minimizes the distance to goal r_i^g; preferring agents closer to the goal. A normalization to this term is applied with the minimum formation distance $\min_{i\in V} r_i^g$.
4. P is a real number which is determined experimentally to keep consistency – when a new agent is better positioned toward the goal, it should outperform the previous agent to change the leader. A penalty cost P is added when the ID_i at time t differs from the past selection at time $t-1$.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

5. $w_{ID}ID_i$ is a term used as a way to select a single agent when there is a tie among more than one agent. ID_i is an integer number indicating the robot identity label.

Regarding the first two terms of J_i^g involving differences of angles, we calculate the minimum angle that a robot needs to rotate to aim the desired orientation.

3.3.2. Goal velocity vector

Once a leader agent is selected, the goal velocity vector p_i^g required in Eq. 17 is determined from the result of the FND. The goal velocity vector is defined as:

$$p_i^g = v_{des}\,\hat{n}_i^g \tag{19}$$

where $v_{des} \in \mathbb{R}$ is a desired velocity magnitude and $\hat{n}_i^g$ is a unit vector with the direction to goal derived from the solution sector with direction φ_θ given by $\hat{n}_i^g = [\cos\varphi_\theta, \sin\varphi_\theta]^T$.

4. Simulation experiments

The proposed flocking method was detailed for application on agents moving in two dimensions; however, it is shown later that the method can be applied in three dimensions with few changes by replacing the ND planner with a planner based on artificial potential fields [26]. First, in this section, we present results on wheeled mobile robots. The method is implemented in the Gazebo simulator using Robotics Operating System (ROS) functionalities and relying only on measurements from onboard sensors. Our approach is quite fast; the control loop runs at 20 Hz, and the implementation is distributed. All the algorithms have been implemented, even the processing of measurements given by the onboard range finders and the communication between the agents.

4.1. Model of agents

We consider differential-drive robots that are modeled by the unicycle kinematics:

$$\begin{bmatrix}\dot{x}_i\\ \dot{y}_i\\ \dot{\theta}_i\end{bmatrix} = \begin{bmatrix}\cos\theta_i & 0\\ \sin\theta_i & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v_i\\ \omega_i\end{bmatrix} \tag{20}$$

where x_i, y_i are the position coordinates of the robot's center and θ_i its heading in a global reference frame, v_i and ω_i are its linear and angular velocities, respectively. To have a model as in Eq. 2, we use a displaced control point described as:

$$\begin{bmatrix}x_i^c\\ y_i^c\end{bmatrix} = \begin{bmatrix}x_i + l\cos\theta_i\\ y_i + l\sin\theta_i\end{bmatrix} \tag{21}$$

with $l \neq 0$ a distance along the longitudinal axis of the robot.

The kinematics of the displaced control point are given by:

$$\dot{q}_i = M_i\begin{bmatrix}v_i\\ \omega_i\end{bmatrix} = p_i \tag{22}$$

where $M_i = \begin{bmatrix}\cos\theta_i & -l\sin\theta_i\\ \sin\theta_i & l\cos\theta_i\end{bmatrix}$. Then, the vector of velocity inputs $w_i = [v_i, \omega_i]^T$ are obtained as:

$$w_i = M_i^{-1}p_i. \tag{23}$$

Since our flocking control scheme provides the acceleration u_i, according to Eq. 2, p_i can be computed by numerical integration of $\dot{p}_i = u_i$ in order to get the robot's velocity inputs.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

Table I. *Parameters used in the simulation results.*

Parameter	Description	Value
r_{α}^{max}	Max distance neighbor detection	5 m
r_{β}^{max}	Max distance obstacle detection	2 m
r_{α}^{des}	Desired inter-agent distance	1 m
r_{β}^{des}	Desired distance from obstacles	1 m
η	Distance discontinuity	0.2 m
v_{des}	Desired velocity	0.15 m/s
w_{θ_g}, w_{θ}, w_r, w_{ID}, P	Cost function weights	6, 5, 1.5, 2, 5

For simulation purposes, we use as agents the robots TurtleBot3, equipped with a 360-degree LiDAR (range of 5 m and resolution of 0.5 deg) and an Inertial Measurement Unit (IMU) to provide velocity and orientation estimation. These robots are small size, diameter $r_{\alpha}^{size} = 0.13$ m, and we consider maximum linear velocity $v_{max} = 0.22$ m/s and maximum angular velocity $\omega_{max} = 0.8$ rad/s. To define the displaced control point, we set $l = 0.05$ m, and the velocity inputs are computed every $\Delta t = 0.05$ s.

4.2. Parameters of the flocking algorithm

The proposed approach uses some parameters, which are listed in Table I.

4.3. Performance metrics and α-flocking

The performance metrics used to evaluate the approach and to compare it with other related works are four:

1. Algebraic connectivity $\lambda_2(t)$: It is the second-smallest eigenvalue of the Laplacian matrix L obtained from the interaction matrix [27].
2. Deviation energy $E(t)$: It is a measure of the deviation of the formation from the α-lattice, determined by the following equation:

$$E(t) = \frac{1}{(|\epsilon_{\alpha D}(t)| + 1)} \sum_{i=1}^{n} \sum_{N_i^{\alpha D}} \kappa_g^{\alpha} \psi_{\alpha}(d_{ij}) \tag{24}$$

   where $\psi_{\alpha}(d_{ij})$ is the integral of the action function $\phi_{\alpha}(d_{ij})$ in Eq. 12, $|\epsilon_{\alpha D}(t)|$ is the size of the α-Delaunay graph. The gain κ_g^{α} used in Eq. 12 is added here to consider the real potential energy applied to the control law.
3. Cohesion radius $R(t)$: It is defined by the following expression: $\max_{i \in \mathcal{V}} ||q_i(t) - q_c(t)||$, where $q_c(t) = \frac{1}{n} \sum_{i=1}^{n} q_i$.
4. Velocity mismatch $K(t)$: It is defined by the following equation:

$$K(t) = \frac{1}{n} \sum_{i=1}^{n} ||p_i - p_c||^2 \tag{25}$$

   where $p_c(t) = \frac{1}{n} \sum_{i=1}^{n} p_i$.

To verify that our algorithm generates a flocking formation, we will use the definition of an α-flocking presented in the Olfati-Saber's approach [8], which is based on the performance metrics presented above, it states: A group of n agents perform α-Flocking over the time interval $[t_0, t_f]$, if there exist relatively small numbers $\epsilon_0, \epsilon_1, \epsilon_2 > 0$ and distance $d > 0$ such that the trajectory satisfies all the following conditions for all $t \in [t_0, t_f]$, with an interaction range $r = (1 + \epsilon_0)d$: (1) The group remains a quasi-flock. (2) The group remains cohesive. (3) The deviation energy remains small $E(t) \leq \epsilon_1 d^2$. (4) The velocity

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

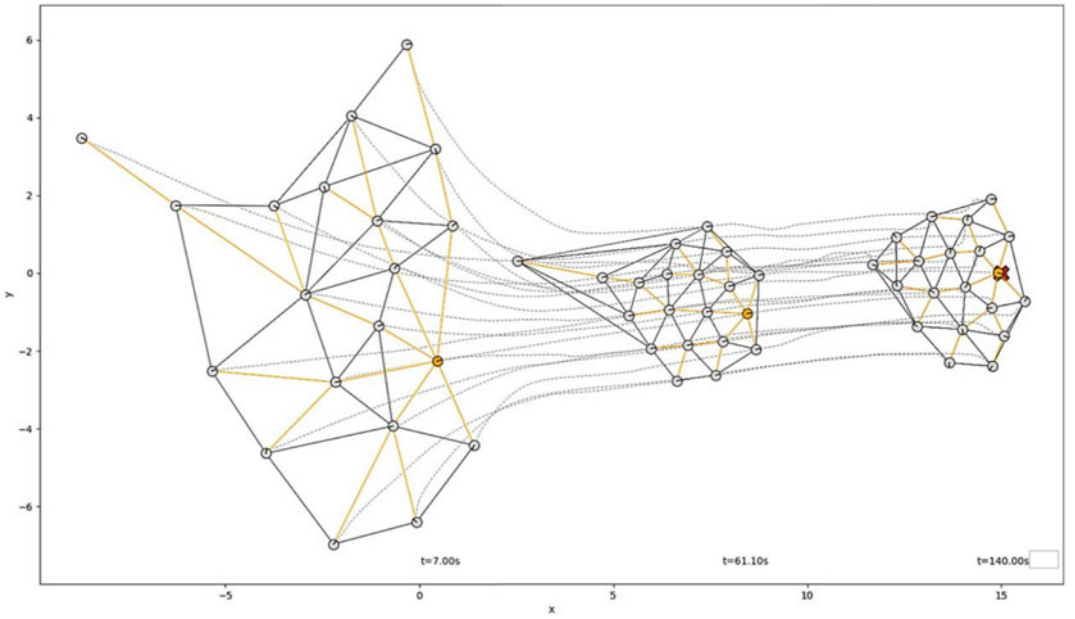


Figure 8. *Simulation snapshots for 20 agents in free space.*

mismatch remains small $K(t) \leq \epsilon_1 n$. In this work, the variables r and d are referenced as $r = r_\alpha^{\max}$ and $d = r_\alpha^{des}$, which are the interaction range and desired inter-agents' distance, respectively.

There are two main differences of our method compared with Olfati-Saber's approach: (1) the proposed method includes a motion planner to establish a goal for the global leader, and (2) it uses local leaders that move the agents toward them. Those innovations are the main contributions of our work. Furthermore, the proposed approach offers improvements compared with Olfati-Saber's model. The proposed approach is able to deal with non-convex obstacles, and it is able to deform the formation to pass through narrow passages. These capabilities are clearly shown in the simulation results presented in the next sections.

4.4. Free space

The algorithm implementation is first tested in free space with $n = 20$ agents, with random initial positions and headings for each agent. The agent's velocity at start time is zero. The goal position provided to the agents is the position $q_g = (0, 15)$ meters.

The evolution of the formation is shown in Figure 8 through snapshots at different simulation times. The leader agent is highlighted in yellow. The edges of the α-Delaunay graph are depicted with black lines connecting neighboring agents. The γ_i agent relationship is indicated with a yellow line. The leader agent is chosen using Alg. 1, applying the cost function specified in Eq. 18.

The control inputs provided to each agent can be visualized in Figure 9. In that figure, it is shown that the leader agent converges to the desired velocity v_{des}; however, the leader is collaborative and adapts its velocity due to the lattice control u_i^α term.

The follower agents increase their velocity until they reach the desired inter-agent's distance with their α-neighbors and γ-agent. Then, they decrease their velocity to match their γ-agent's velocity, which is the chain formation consensus velocity v_{des}. As can be seen in Figure 9, the leader reaches the goal around $112s$, it stops, and the velocity of the followers converges around zero.

Figure 10 shows graphics of the performance metrics described in Section 4.3. The first clear difference from the Olfati-Saber's framework [8] is that our interaction range is not bounded by the restriction $r = (1 + \epsilon_0)d$. Our formation fulfills the α-flocking for any values of ϵ_0, as long as condition $r_\alpha^{des} < r_\alpha^{\max}$

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

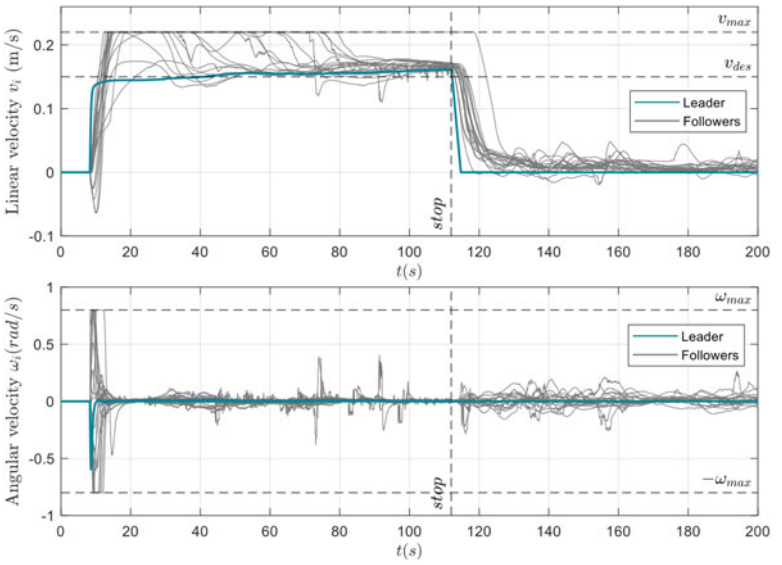


Figure 9. *Velocity inputs for the agents in simulation in Figure 8.*

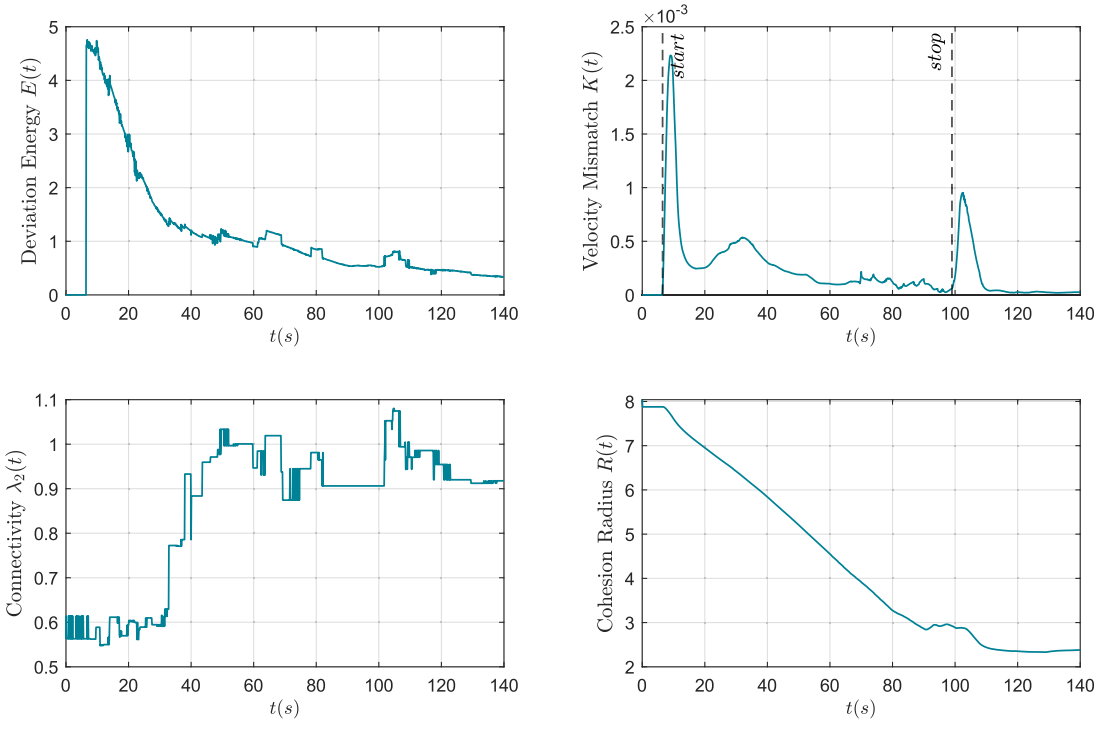


Figure 10. *Performance metrics for the simulation of 20 agents in free space.*

is fulfilled. In our simulation configuration, the interaction range ratio is $\epsilon_0 = 4$, while in Olfati-Saber's work, a low ratio of $\epsilon_0 = 0.2$ is necessary to fulfill these conditions.

The accomplishment of α-flocking condition 1), referring to the quasi-flock formation, is shown in Figure 10. The connectivity graph $\lambda_2(t)$ increases as the simulation time progresses, and therefore, the graph remains connected for all $t \in [t_0, t_f]$.

The condition 2), which refers to the formation cohesion, is shown to be fulfilled in the plot of the Cohesion radius $R(t)$, where the formation radius decreases to a low value.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

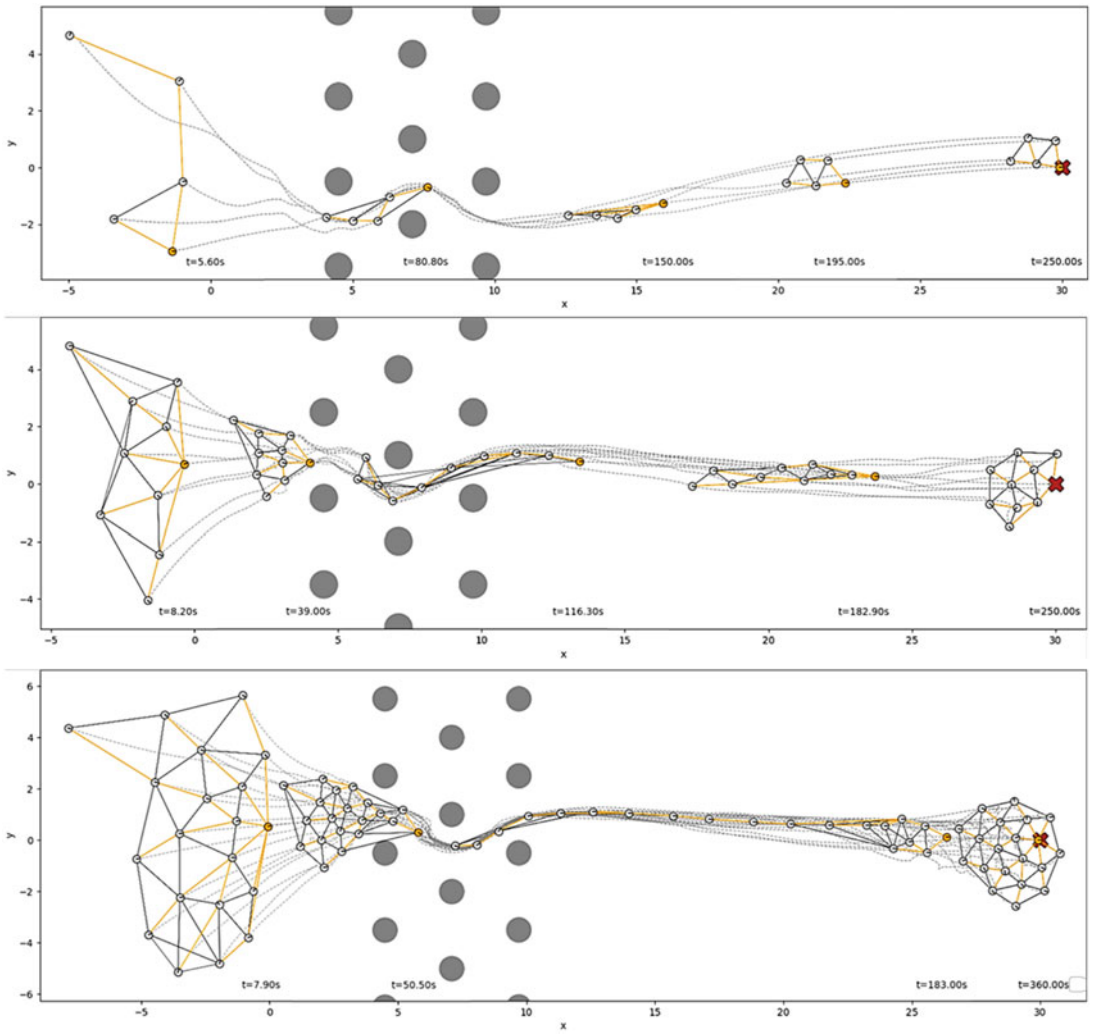


Figure 11. *Snapshots for $n = 5,10$, and 20 agents in an environment with $d_{obs} = 2$.*

For the last two conditions for α-flocking, it is necessary to define a value for ϵ_1 and ϵ_2 constants. These constants are not bounded to any specific value; they are only required to be relatively small numbers. Figure 10 shows the evolution of energy deviation $E(t)$ and velocity mismatch $K(t)$.

The energy deviation values at start and end time are $E(t_0) \approx 7.375$ and $E(t_f) \approx 0.176$, respectively. These values give the constant ϵ_1 a value of $\epsilon_1 = 0.176$ or higher to fulfill condition 3). It can be noticed that the constant is much lower than the highest energy at the start time $0.176 \ll 7.375$.

The velocity mismatch values at start and end time are $K(t_0) \approx 2.23 \times 10^{-5}$ and $K(t_f) \approx 5.74 \times 10^{-6}$, respectively, and it reaches a maximum of $K(t = 10.2s) \approx 1.732 \times 10^{-3}$ (see Figure 10 top-right). These simulation values suggest that a constant value $\epsilon_2 = 2.87 \times 10^{-7}$ or higher is required to satisfy condition 4). It can be seen that the constant is much lower than the highest velocity mismatch reached at $t = 10.2s$, $2.87 \times 10^{-7} \ll 1.732 \times 10^{-3}$.

Considering the selection of constants ϵ_0,ϵ_1,ϵ_2, it can be concluded that the algorithms implemented accomplished the α-flocking definition proposed in ref. [8].

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

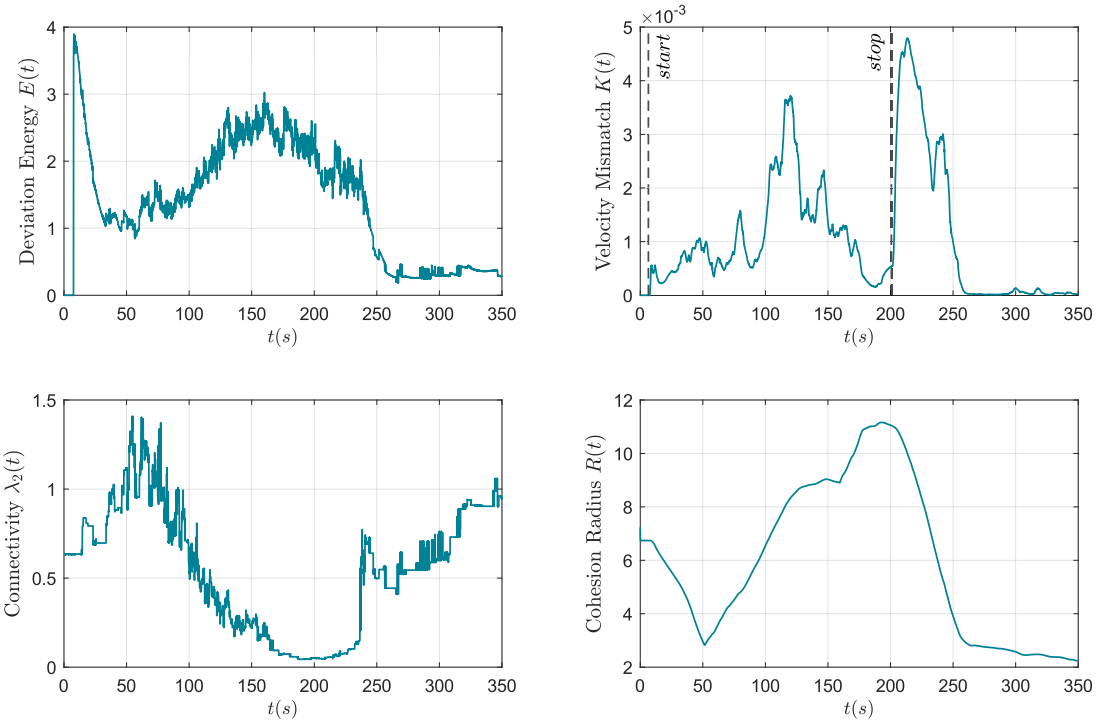


Figure 12. *Metrics for the $n = 20$ and $d = 2$.*

4.5. Convex obstacles

An environment was created in Gazebo where the obstacles are cylinders with a radius of 0.5 m and free space between the cylinders $d_{obs} = 2$ meters. The desired distance from the obstacle is set to $r_{\beta}^{des} = 1$ meter. This scenario was simulated 10 times for $n = 5$, 10, and 20 agents, where each agent starts at a random initial position in the range $x_i \in [-10, 0]$ and $y_i \in [-10, 10]$, and random initial headings. The initial velocity is zero for all agents. The goal position is set to $q_g = (30, 0)$ for all simulations.

Figure 11 shows a snapshot of a simulation. In that figure, one can observe a particular behavior of the agents; the formation is transformed to a chain-like shape to be able to avoid collision with the obstacles in the clutter environment. This behavior gives the name of *chain-flocking* to this method.

The effect of chain formation can also be seen in Figure 12. The graph connectivity λ_2 is visibly reduced when the agents traverse the narrow passage formed by the obstacles due to the loss of edge connections needed to form the chain. This also causes an increment in the formation cohesion radius. The deviation energy and velocity mismatch do not decrease until the obstacles are surpassed, as the agents deviate from their trajectories to form the lattice to be able to avoid collisions with obstacles.

In the 10 experimental runs realized for 5, 10, and 20 agents, the minimum inter-agent's distances were 0.49, 0.42, and 0.4 m, respectively, while the minimum agent-obstacle distances were 0.59, 0.51, and 0.5 m, respectively. This shows that no collisions occur with other agents or obstacles.

4.6. Non-convex complex environment: Spiral-shaped obstacle

The proposed approach was also tested in a spiral-shaped environment, where there is only one exit path toward the goal position. This environment is illustrated in Figure 13.

The simulation is performed for $n = 15$ agents, and two cases are presented; in the first case, the agents are coming out of the spiral obstacle with the goal target set to $q_g = (20, 0)$, and in the second case, the agents are commanded to reach the target goal $q_g = (0, -3)$, which is at the interior of the

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

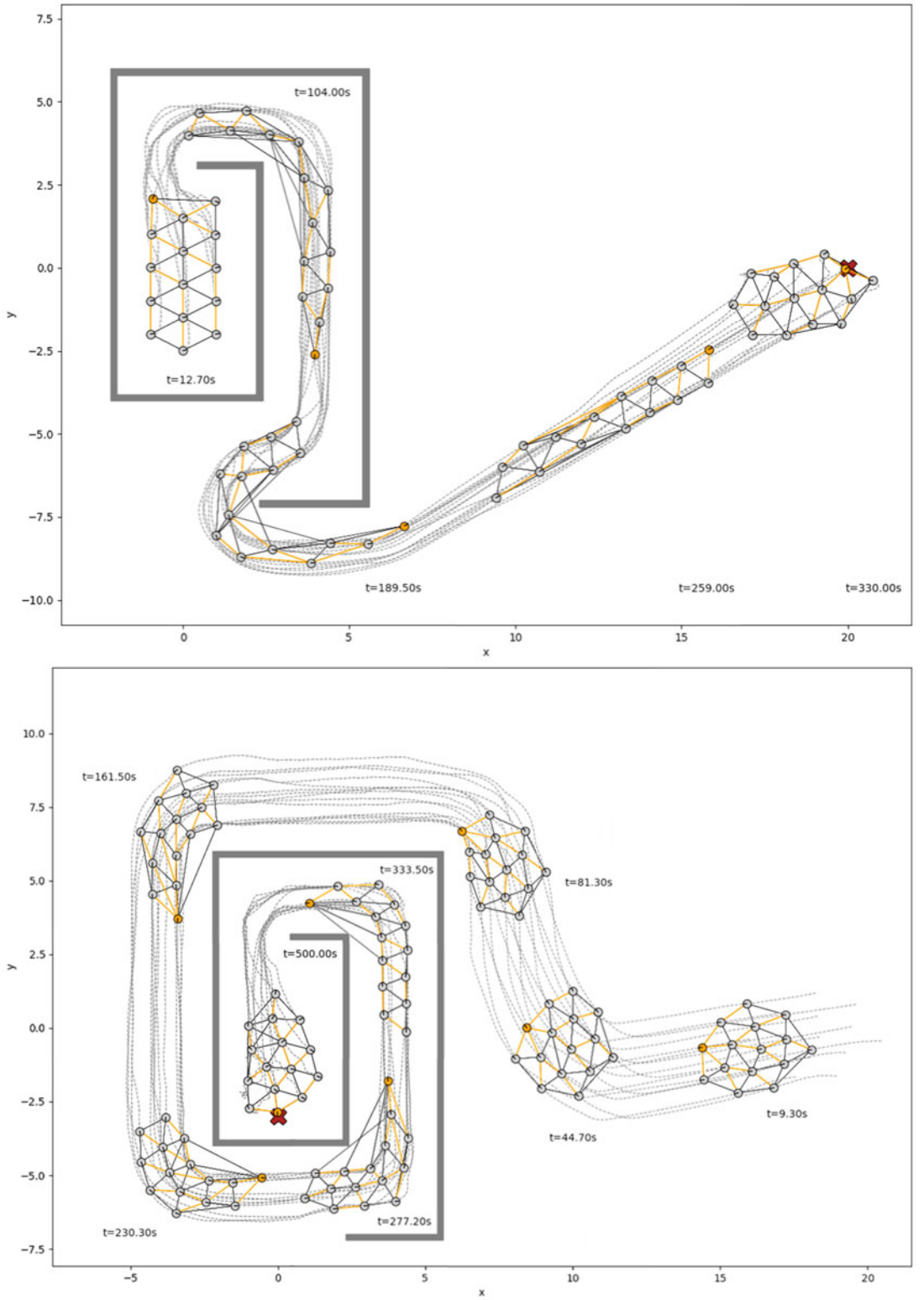


Figure 13. *Snapshots of spiral environment simulations with* $n = 15$ *agents. Up: Getting out of the spiral. Down: Coming back to the spiral.*

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

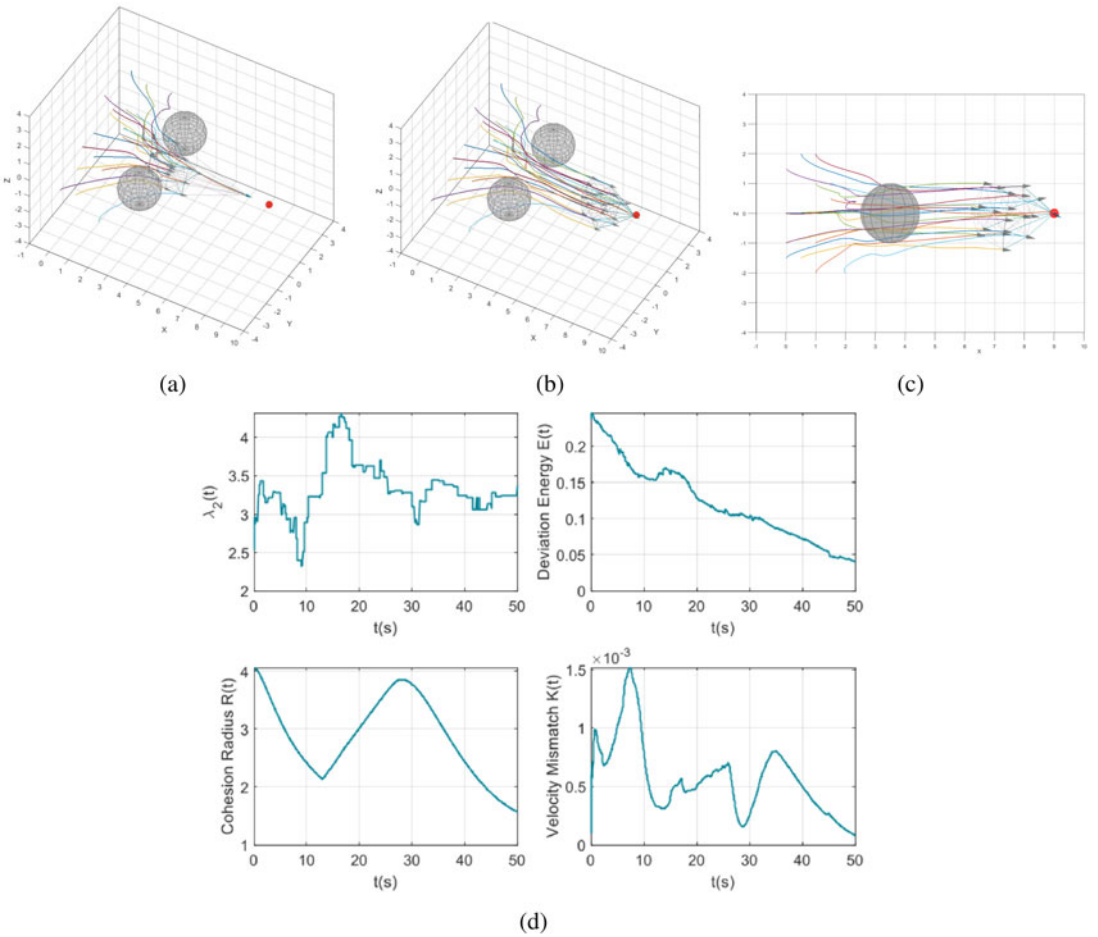


Figure 14. *Snapshots of a simulation in 3D for 20 agents. (a) and (b) show isomorphic views, while (c) shows a side view, and (d) presents plots of the metrics.*

spiral obstacle. Figure 13 shows the formation behavior in this type of obstacle through snapshots of the simulation.

For all simulations, the agents apply the FND motion planning method presented in Section 3.3 to calculate a navigation path that lets them approach to the goal position. However, in this type of environment, the FND characteristics become more evident, where the agents prevent the generation of paths that could trap the agents in local minima, generating only paths at the traversable valleys.

4.7. Implementation in 3D

In this section, we present simulation results in a 3D environment implemented in MATLAB for double-integrator agents as in Eq. (2). In this simulation, a potential field method [26] is used as a motion planner instead of the extended version to multiple agents proposed in this work, of the classical ND method [9]. Figure 14(a)-(c) shows snapshots of a simulation in 3D for 20 agents with desired inter-agent distance $d = 1$. Notice that the formation changes its 3D shape to pass through the obstacles. The agents also adapts their altitude (motion in z-axis) to facilitate the navigation (see Figure 14(c)). As can be seen in Figure 14(d), in this simulation, the energy deviation $E(t)$ starts in 0.25 and reaches 0.045, the cohesion radius $R(t)$ starts in 4.05 and reaches 1.6, and the velocity mismatch $K(t)$ at the end is 8×10^{-5}, which shows the good performance of the approach in the 3D case.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

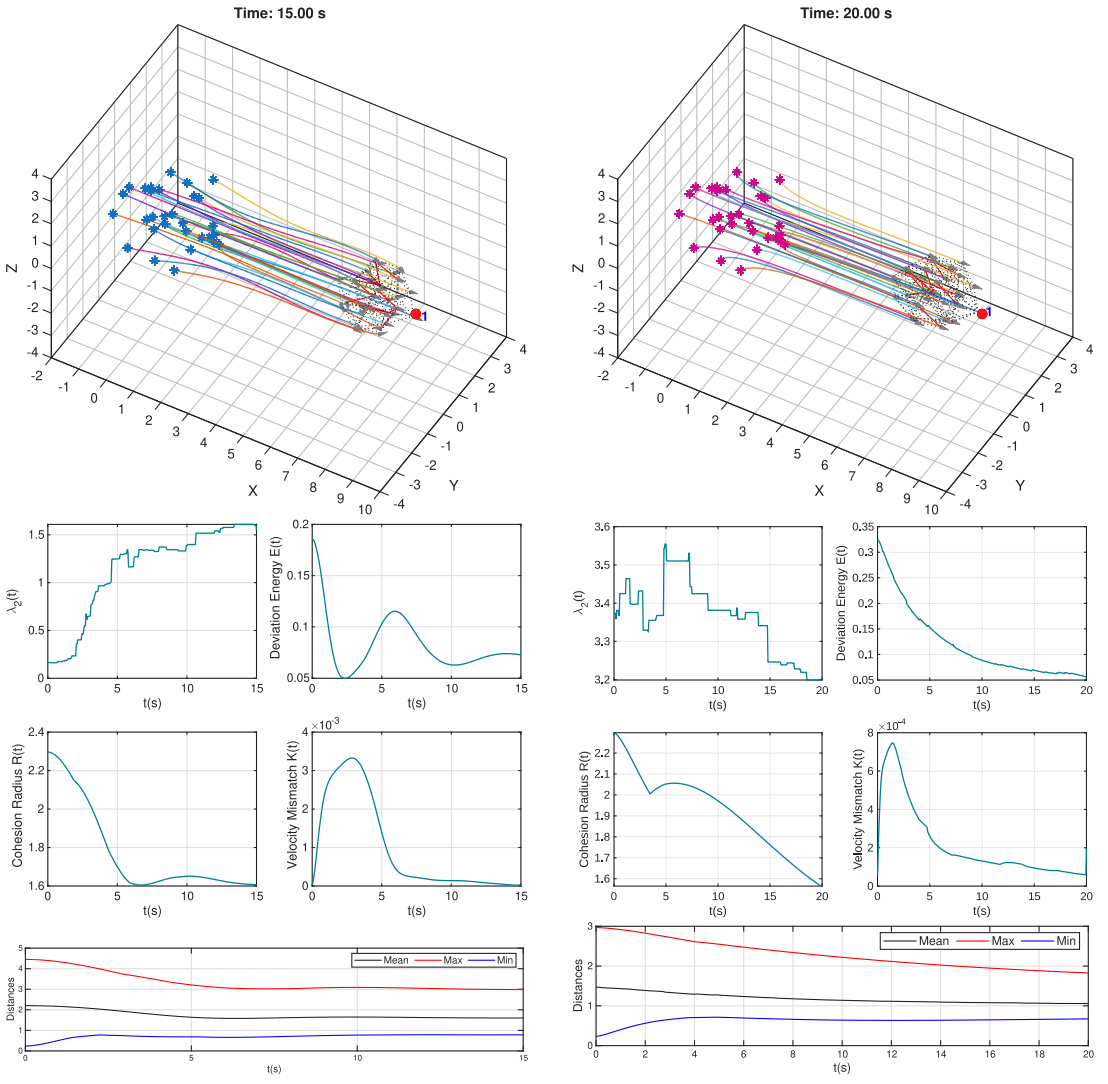


Figure 15. *Comparison between Olfati-Saber's approach [8] (figures in the left column) and the proposed approach (right column) for the same initial conditions in free space.*

In the same 3D implementation, we compared the performance of the Olfati-Saber's approach and the proposed approach, which is shown in Figure 15 for free space. First, the Olfati-Saber's approach achieves flocking only if a small perception range is used, slightly larger than the desired inter-agents distance, here $r_\alpha^{max} = 1.2$ while $r_\alpha^{des} = 1$. In our approach, thanks to the modeling of connectivity, we can use a realistic perception range $r_\alpha^{max} = 5$. This allows us to solve the flocking for sparse initial positions of the agents; it can be seen in the λ_2 metric that Olfati-Saber's approach starts with small values (poor connectivity) and increases as time elapses, whereas in our approach, the connectivity values start high and remain so during evolution. Our approach reaches a smaller energy deviation $E(t)$, a similar cohesion radius $R(t)$, and a smaller velocity mismatch $K(t)$. The last plots in each column of Figure 15 present the mean, maximum, and minimum distances among agents over the time of the simulation. They show that our approach at the end time reaches a mean distance very close to the desired value $r_\alpha^{des} = 1$, and the maximum and minimum distances are also closer to that value than in the Olfati-Saber's approach.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

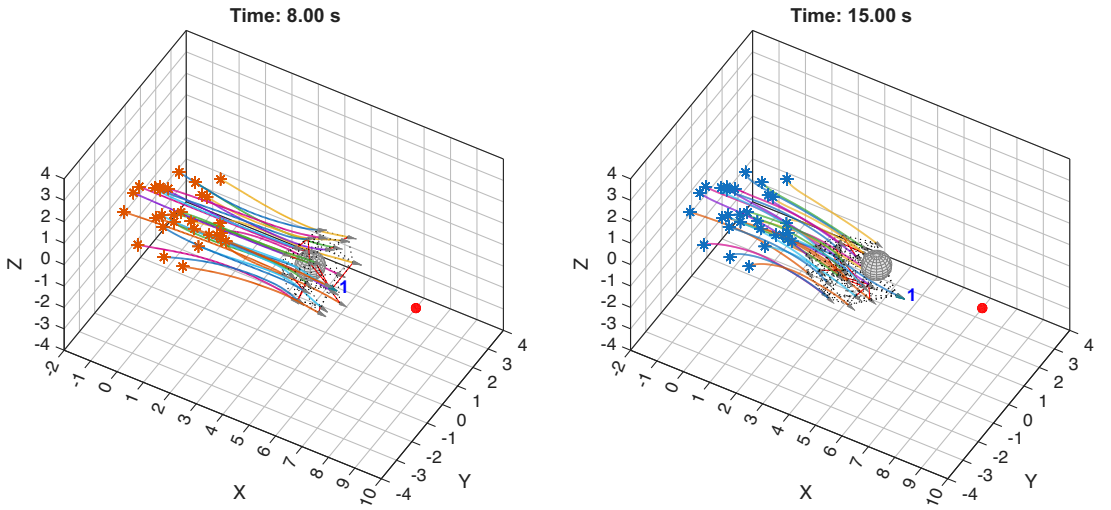


Figure 16. *Comparison between Olfati-Saber's approach [8] (left) and the proposed approach (right) evading an obstacle.*

Finally, in Figure 16, we present the advantage of our approach to deal with an obstacle placed exactly on the path to the goal. Whereas in the Olfati-Saber's approach, the agents evade the obstacle by spreading out around it (which causes disconnection between agents), with our approach, all the agents evade the obstacle by passing underneath it, keeping connectivity.

Thus, our method offers advantages that the Olfati–Saber's model lacks. First, the sensor range influences the neighborhood created; a large range implies connectivity between all agents, which causes inconsistencies in meeting the distance constraint between agents. Some agents cannot be simultaneously close to all their neighbors, which is avoided in our approach. Second, the ability to deform the formation in the presence of obstacles prevents the appearance of isolated groups that do not communicate with each other, which does occur in the Olfati–Saber's model. Furthermore, the proposed method frequently avoids getting trapped in local minima generated by non-convex obstacles, which frequently happens with the Olfati–Saber's model.

We invite the reader to watch a video with simulations similar to those shown in this section: https://n9.cl/74fy7.

5. Conclusions and future work

In this article, we have proposed a navigation approach based on flocking algorithms for multiple robots. The proposed approach seeks to move the agents in a flexible formation that preserves a certain pattern that minimizes a measure of energy, but is also able to avoid collision with obstacles while maintaining communication and visibility between agents. Unlike other works, the location of the robots in a global framework is unknown; only a global direction of motion is known. The location of obstacles in the environment is also unknown a priori; the robots discover obstacles as they move and calculate their motion velocities using measurements obtained with sensors onboard the robots.

Gradient-based control laws are combined with a new version of the planner called the ND, extended as FND. The task of this local planner is to select motion directions that favor reaching the goal, while avoiding collisions between agents and avoiding obstacles of general shapes, including concave ones. Two central ideas in the approach are: (1) the global leader of the formation can change, selecting as global leader one that has favorable conditions to reach the goal, (2) each agent locally follows another agent, which we call chain-flocking; the agent that is followed minimizes a cost related to the shortest path to the global leader in a graph.

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

The practical result is an approach that frequently maintains communication formation, forming an agent graph that has only one connected component and that is flexible enough to allow agents to line up when narrow passages appear in the environment. The proposed method is intensively validated with several computer simulations, implemented in particular for differential-drive robots.

For future work, we plan to implement the proposed approach on real robots. Besides, we consider proposing controllers with finite-time stability in order to have faster convergence and finish the navigation task in a shorter time.

Author contributions. M. R. worked in validation, software, and formal analysis. H. M. B. and R. M. C. worked in writing, reviewing, methodology, formal analysis, and conceptualization.

Financial support. The first author was granted by the National Council for the Humanities, Science, and Technology (CONAHCYT), number 1141708.

Competing interests. The authors declare none.

Ethical approval. Not applicable.

References

[1] K. K. Oh, M. C. Park and H. S. Ahn, "A survey of multi-agent formation control," *Automatica* **53**, 424–440 (2015).
[2] P. Shi and B. Yan, "A survey on intelligent control for multiagent systems," *IEEE Trans. Syst. Man Cybern. Syst.* **51**(1), 161–175 (2021).
[3] A. Okubo, "Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds," *Adv. Biophys.* **22**, 1–94 (1986).
[4] D. J. T. Sumpter, "The principles of collective animal behaviour," *Phil. Trans. R. Soc.* **361**(1465), 5–22 (2006).
[5] T. A. Collins, B. M. Yeoman and P. Katira, "To lead or to herd: Optimal strategies for 3D collective migration of cell clusters," *Biomech. Model. Mechanobiol.* **19**(5), 1551–1564 (2020).
[6] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.* **21**(4), 25–34 (1987).
[7] H. G. Tanner, A. Jadbabaie and G. J. Pappas, "Stable Flocking of Mobile Agents, Part I: Fixed Topology," **In**: *42nd IEEE Int. Conf. on Decision and Control*, vol. 2, IEEE (2003) pp. 2010–2015.
[8] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Automat. Control* **51**(3), 401–420 (2006).
[9] J. Minguez and L. Montano, "Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios," *IEEE Trans. Robot. Autom.* **20**(1), 45–59 (2004).
[10] J. W. Durham and F. Bullo, "Smooth Nearness-Diagram Navigation," **In**: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, IEEE (2008) pp. 690–695.
[11] Y. Cao, W. Yu, W. Ren and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Ind. Informat.* **9**(1), 427–438 (2013).
[12] Q. Wang, H. Fang, J. Chen, Y. Mao and L. Dou, "Flocking with Obstacle Avoidance and Connectivity Maintenance in Multi-Agent Systems," **In**: *IEEE Conf. on Decision and Control (CDC)*, IEEE (2012) pp. 4009–4014.
[13] W. Zhao, H. Chu, M. Zhang, T. Sun and L. Guo, "Flocking control of fixed-wing UAVs with cooperative obstacle avoidance capability," *IEEE Access* **7**, 17798–17808 (2019).
[14] W. Wu, X. Zhang and Y. Miao, "Starling-behavior-inspired flocking control of fixed-wing unmanned aerial vehicle swarm in complex environments with dynamic obstacles," *Biomimetics* **7**(4), 214 (2022).
[15] Y. Zhao, Z. Qu, H. Liu and R. Zhu, "Bio-inspired robot swarm path formation with local sensor scope," *Appl. Intell.* **53**(14), 17310–17326 (2023).
[16] J. Yu, J. Ji, Z. Miao and J. Zhou, "Fully distributed region-reaching control with collision avoidance for multi-robot systems," *Robotica* **39**(8), 1405–1416 (2021).
[17] H. Su, X. Wang and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Trans. Autom. Control* **54**(2), 293–307 (2009).
[18] D. Sakai, H. Fukushima and F. Matsuno, "Flocking for multirobots without distinguishing robots and obstacles," *IEEE Trans. Control Syst. Technol.* **25**(3), 1019–1027 (2017).
[19] J. Wu, Y. Ji, X. Sun, W. Fu and S. Zhao, "Anonymous flocking with obstacle avoidance via the position of obstacle boundary point," *IEEE Internet Things J.* **12**(2), 2002–2013 (2025).
[20] D. Sakai, H. Fukushima and F. Matsuno, "Leader–follower navigation in obstacle environments while preserving connectivity without data transmission," *IEEE Trans. Control Syst. Technol.* **26**(4), 1233–1248 (2018).
[21] X. Li, R. Zhou, G. Sun and J. Zhang, "Connectivity-preserving flocking of multiagent systems via selecting critical neighbors," *IEEE Trans. Netw. Sci. Eng.* **10**(6), 3779–3792 (2023).

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press

[22] Y. Huang, W. Fang, Z. Chen, Y. Li and C. Yang, "Flocking of multiagent systems with nonuniform and nonconvex input constraints," *IEEE Trans. Autom. Control* **68**(7), 4329–4335 (2023).
[23] R. Klein, *Voronoi Diagrams and Delaunay Triangulations* (Springer, New York, NY, 2016).
[24] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks* (Princeton Series in Applied Mathematics, 1 edition, (Princeton University Press, 2010).
[25] H. Edelsbrunner, D. Kirkpatrick and R. Seidel, "On the shape of a set of points in the plane," *IEEE Trans. Inf. Theory* **29**(4), 551–559 (1983).
[26] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *IEEE Int. Conf. Robot. Autom.* **2**, 500–505 (1985).
[27] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Control* **49**(9), 1520–1533 (2004).

Cite this article: M. Rodriguez-Ortiz, H. M. Becerra and R. Murrieta-Cid (2026). "Chain-flocking: navigation of multiple agents with local measurements in unknown environments", Robotica **44**, 1135–1158. https://doi.org/10.1017/S0263574726103506

https://doi.org/10.1017/S0263574726103506 Published online by Cambridge University Press