Robotica (2015) volume 33, pp. 332–347. © Cambridge University Press 2014
doi:10.1017/S0263574714000368

Edge-weighted consensus-based formation control strategy with collision avoidance

Riccardo Falconi†, Lorenzo Sabattini‡*, Cristian Secchi‡, Cesare Fantuzzi‡ and Claudio Melchiorri†

†*Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi" (DEI), University of Bologna, Italy*
‡*Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy*

(Accepted January 31, 2014. First published online: February 28, 2014)

SUMMARY
In this paper, a consensus-based control strategy is presented to gather formation for a group of differential-wheeled robots. The formation shape and the avoidance of collisions between robots are obtained by exploiting the properties of weighted graphs. Since mobile robots are supposed to move in unknown environments, the presented approach to multi-robot coordination has been extended in order to include obstacle avoidance. The effectiveness of the proposed control strategy has been demonstrated by means of analytical proofs. Moreover, results of simulations and experiments on real robots are provided for validation purposes.

KEYWORDS: Formation control; Weighted graphs; Collision avoidance.

1. Introduction

This paper introduces a decentralized control strategy to let a group of robots create a desired geometric formation by means of local interaction with neighboring robots.

The idea of studying algorithms to let a group of mobile agents perform formation control has been directly inspired by the behavior of *social animals*,[1,2] where local interactions between agents and simple behavioral control exploited by each of them drives to a complex behavior for the entire system, such as in the case of school of fish or birds flocking.

The emergence of complex *social* behaviors leads to the study of more formalized forms of interactions between agents.[3–6] In particular, it drove the attention to the study of algorithms able to ensure that agents can achieve and then preserve a predefined formation, even in the presence of environmental constraints. Formation control is a very well-studied problem, and many different approaches can be found in the literature. In the centralized (or leader follower) control strategies, the leader (that can either be one of the vehicles, or an external computation unit) computes the control inputs for the followers. Due to the intrinsic unreliability of this class of control strategies,[7] this paper focuses on distributed ones.

A remarkable example of decentralized formation control strategy based on consensus has been introduced in Fax and Murray:[8] In this work, the authors describe how to exploit consensus algorithms to obtain a formation of autonomous vehicles whose interconnection is described by means of a graph. One of the main advantages of consensus algorithms is the fact that the agreement is reached even in the presence of delays in the communication.[9] Furthermore, the multi-agent system keeps a stable behavior even in the case of a variable communication topology.[8]

Since robots are moving in the same environment, collisions among them can occur during the creation of the formation. Furthermore, if the environment is (at least partially) unknown and unstructured, collisions with obstacles can occur as well. Artificial potential fields (see, e.g., refs. [7,

* Corresponding author. E-mail: lorenzo.sabattini@unimore.it

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

10–14], and references therein) are a very effective way to implement collision avoidance strategies. Hence, to include collision avoidance among the agents into consensus-based formation control strategies, typically repulsive potential fields are added.[9] One of the main drawbacks in using potential fields is the fact that delays in the communication channels may drive the system to instability.[15]

The main contribution presented in this paper is a novel approach based on edge-weighted graphs[16,17] introduced in order to define a new behavioral control strategy for a group of mobile robots moving in unknown environments. In particular, edge-weight functions have been appropriately defined to obtain the desired formation. Moreover, a local minima avoidance strategy has been introduced to assure that the robots always converge to the desired configuration. The proposed control framework has been developed in order to guarantee collision avoidance among robots. Moreover, introducing virtual agents and negative edge-weights, the proposed strategy has been extended to include also the possibility to avoid collisions with detected obstacle.[18] The main problem related to obstacle avoidance is that the formation could not be always preserved,[19] e.g., if the robots have to pass through narrow passages. Thus, one of the new improvements introduced in this work focuses on the possibility of creating complex behaviors by dynamically changing only few parameters of the weight functions describing inter-robot relationships. For example, the control algorithm allows the robots to change the formation rigidity depending on environmental constraints.[20] Moreover, since the algorithm defined in this paper is fully consensus-based, it is robust to the presence of communication delays. This paper extends the results preliminary presented in Falconi *et al.*[21]

The paper is structured as follows. In Section 2, background notions about graph theory and consensus algorithms are summarized; in Section 3 the proposed consensus-based algorithm is introduced and the system convergence and stability are supported by means of mathematical proof, focusing, in particular, on the possibility to exploit this approach to avoid collisions among agents. The same approach is then used in Section 4 to perform obstacle avoidance in the case of robots moving in unknown environments. Simulation results are presented in Sections 6.1 and 6.2, and experimental data are collected and analyzed in Section 6.3. In Section 7, conclusions and future works are summarized.

2. Preliminaries

2.1. Background on graph theory

In this section, some of the main graph theory results are summarized. Further details can be found, for instance, in Royle and Godsil.[22] Suppose that the communication topology among the group of robots can be described by means of an undirected graph. An undirected graph with N elements is defined as a pair $G = (V, E)$, where

- $V = \{v_i, i = 1 \dots N\}$ is the *vertex set*,
- $E \subseteq V \times V$ is the *edge set*.

Since in this paper only undirected graphs will be considered, it follows that the elements of E are unordered pairs of elements, namely:

$$(v_i, v_j) \in E \Leftrightarrow (v_j, v_i) \in E.$$

The *Neighbors subset* of the ith agent is defined as

$$\mathcal{N}_i = \{\forall v_j \in V : (v_i, v_j) \in E\}.$$

Each agent is supposed to be able to exchange data with its neighbors, that is, with all the agents that are in its Neighbors subset.

Given a graph $G = (V, E)$ and an orientation map defined over the edge set, the *Incidence matrix,* $\mathcal{I} \in \mathbb{R}^{N \times M}$, can be defined as:

$$i_{i,k} = \begin{cases} -1 & \text{if } \varepsilon_k = (v_i, v_j) \\ 1 & \text{if } \varepsilon_k = (v_j, v_i), \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

where M is the cardinality of the edge set and ε_k is the kth edge of G. When the orientation map is not defined, a random orientation can be chosen.

From the definition of the Incidence matrix, a definition of the *Laplacian matrix* follows, which highlights the edges present in a graph:

$$\mathcal{L} = \mathcal{I} \cdot \mathcal{I}^T. \tag{2}$$

The Laplacian matrix has some following remarkable properties:[22]

1. $\mathcal{L}$ is positive semi-definite.
2. The eigenvalues of the Laplacian matrix are always non-negative. Moreover, they can always be ordered as follows:

$$eig(\mathcal{L}) = \{0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N\}.$$

3. If λ_1 is a simple eigenvalue (i.e. $0 = \lambda_1 < \lambda_2$), then the graph is connected. Moreover, in this case

$$\text{null}(\mathcal{L}) = \text{span}\{\mathbf{1}\},$$

where $\mathbf{1} = [1 \ldots 1]^T$ and $\mathbf{0} = [0 \ldots 0]^T$ are vectors of N elements all equal to 1 and 0 respectively. This implies $\mathcal{L}\mathbf{1} = \mathbf{0}$.

For further details on algebraic graph theory, the reader is referred to Royle and Godsil.[22]

2.2. *Background on consensus problem*

The *consensus problem*[23] is a well-known and widely studied problem in the field of decentralized control. It starts from considering all the agents as holonomic kinematic models:

$$\dot{x}_i = u_i, \tag{3}$$

where $x_i \in \mathbb{R}$ is the state of the ith agent. The consensus problem for N agents, whose goal is to drive the whole system to a final common state, can be solved, in a completely decentralized way, with the Laplacian-based feedback method. The feedback control is in the form

$$\dot{x}_i = -\sum_{j \in \mathcal{N}_i} w_{ij}(x)(x_i - x_j), \tag{4}$$

where $x = [x_1, \ldots, x_N]^T$, and $w_{ij}(x)$ are positive edge weight functions. It is worth noting that the edge weights w_{ij} used in this approach are only function of x_i and x_j, thus implementing a fully decentralized algorithm. However, the edge weight function will be referred to as $w_{ij}(x)$ for ease of notation. Let the *Weight matrix* $\mathcal{W}(z)$ be defined as follows:

$$\mathcal{W}(x) = \text{diag}(\{w_{ij}(x) \; s.\, t.\, (v_i, v_j) \in E\}) \in \mathbb{R}^{M \times M}. \tag{5}$$

Furthermore, let the weighted Laplacian matrix be defined as follows:

$$\mathcal{L}_{\mathcal{W}}(x) = \mathcal{I} \cdot \mathcal{W}(x) \cdot \mathcal{I}^T. \tag{6}$$

The control law in Eq. (4) can be rewritten in the following matrix form:[16]

$$\dot{x} = -\mathcal{L}_{\mathcal{W}} x. \tag{7}$$

For further details of consensus problem, the reader is referred to ref. [23].

So far, only scalar states have been taken into account. Consider now the position of each agent as its own state. More specifically, if the position of the ith agent is n-dimensional, the ith agent's state is given by $x_i = [x_{i,1}, \ldots, x_{i,n}]^T$. Considering N agents, the state of a multi-agent system may be defined as $x = [x_1^T, \ldots, x_N^T]^T$.

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

However, for the sake of clarity, the dynamics of the system will be hereafter considered in one dimension only, without loss of generality. All the results presented, in fact, may be easily extended to the n-dimensional case.

3. Control Law for Formation Achieving

The consensus-based formation control algorithm introduced in ref. [8] exploits the following control law:

$$\dot{x} = -\mathcal{L}x + b, \tag{8}$$

where b is defined as a *bias* vector. Let $\bar{x}$ be the vector that contains the desired positions for the agents with respect to the centroid of the formation: the bias vector is defined as $b = -\mathcal{L}\bar{x}$. One of the main drawbacks of the control law in Eq. (8) is that the definition of the formation by means of the desired positions described by the vector $\bar{x}$ does not allow the formation to rotate and adapt its shape to avoid collisions while moving in unknown environments. In fact, the vector $\bar{x}$ defines the positions that each agent have to match in each dimension with respect to the centroid of the group,[8] and thus it must be recalculated for each of these with respect to a world fixed frame. Alternatively, it will be shown that given an appropriate choice of edge-weights $w_{ij}(x),\ \forall\ (v_i, v_j) \in E$, it is possible to obtain the desired formation without using a bias in the control law. In fact, the weight functions may be opportunely defined in order to make the inter-robot distances converge to a desired values, thus creating the desired formation. Furthermore, in the control law presented in this paper, the weight functions can be chosen such that collisions among the agents are always avoided. More specifically, it will be proven that, given a safety distance δ, if the initial configuration of the system is such that all the inter-agent distances are strictly greater than δ, then they will never go below this value.

In ref. [16] edge-weight functions $w_{ij}(x)$ were designed for ensuring the preservation of the connectivity of the graph, defining the control law that prevented edges of the communication graphs from being disconnected. In this paper, inspired by Ji and Egerstedt,[16] the edge-weight functions are designed with the purpose of obtaining the desired formation while avoiding inter-robot collisions. This objective is obtained exploiting the control law introduced in Eq. (4).

For this purpose, let l_{ij} be the edge vector between agents i and j, i.e. $l_{ij} = x_i - x_j$. Furthermore, define a collision-free realization of graph G as

$$\mathcal{D}^{\epsilon}_{G,\delta} = \left\{ x \in \mathbb{R}^{nN} :\ (\delta + \epsilon) \leq \left\| l_{ij} \right\| \leq D_M, \forall (v_i, v_j) \in E \right\} \tag{9}$$

for some positive ϵ, where D_M is the maximum allowed distance that guarantees connectivity between the agents, and δ is the safety distance, that is, the minimum distance that guarantees collision avoidance.

Then an edge-tension function V_{ij} is defined as follows (Fig. 1(a)):

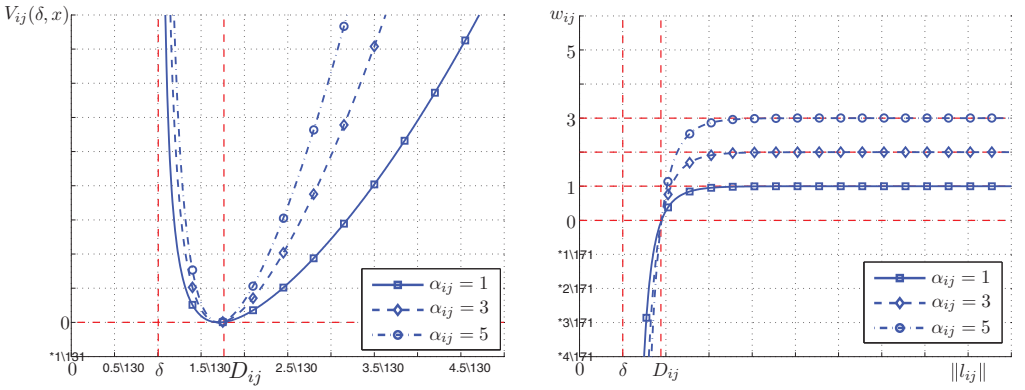
$$V_{ij}(\delta, x) = \begin{cases} \alpha_{ij} \left(\coth\left(\dfrac{\left\| l_{ij} \right\| - \delta}{K_{ij}} \right) + \dfrac{1}{2} \left\| l_{ij} \right\|^2 - V_{ij}^{\min} \right) & \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } (v_p, v_v) \in E, \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{10}$$

where $K_{ij} > 0$ is a constant, $\alpha_{ij} > 0$ is a value used to define the *intensity* of the inter-robot influence,[20] and $V_{ij}^{\min} > 0$ is defined such that

$$\min_{\|l_{ij}\| > \delta} V_{ij}(\delta, x) = 0. \tag{11}$$

It is worth noting that, according to its definition in Eq. (10), V_{ij} is not a function of the entire state x. In fact, since $l_{ij} = x_i - x_j$, V_{ij} is a function of only x_i and x_j. However, it will be hereafter referred to as $V_{ij}(\delta, x)$ for ease of notation.

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press



(a) Edge-tension function $V_{ij}(\delta, x)$ with respect to $\|l_{ij}\|$.

(b) Edge-weight function with respect to $\|l_{ij}\|$.

Fig. 1. Values of the edge-tension function $V_{ij}(\delta, x)$ and the edge-weight function with respect to $\|l_{ij}\|$.

This function is non-negative, and has a strict minimum in $\|l_{ij}\| = D_{ij}$, with

$$D_{ij} = \delta + K_{ij}\text{acsch}(\sqrt{K_{ij}}), \tag{12}$$

where acsch $(\cdot)$ is the inverse function of csch $(\cdot)$. The choice of the value of the constant $K_{ij} > 0$ is related to the position of the minimum of the edge-tension function, i.e. the desired distance for each couple of agents. From Eq. (10), it follows that

$$\frac{\partial V_{ij}(\delta, x)}{\partial x_i} = \begin{cases} \alpha_{ij}\left[-\text{csch}^2\left(\dfrac{\|l_{ij}\| - \delta}{K_{ij}}\right) \cdot \dfrac{(x_i - x_j)}{K_{ij}\|l_{ij}\|} + (x_i - x_j)\right] & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}. \tag{13}$$

The total tension energy of graph G can then be defined as follows:

$$V(\delta, x) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} V_{ij}(\delta, x). \tag{14}$$

Moreover, the edge-weights are defined as follows (Fig. 1(b)):

$$w_{ij} = \alpha_{ij}\left(-\frac{1}{K_{ij}\|l_{ij}\|}\text{csch}^2\left(\frac{\|l_{ij}\| - \delta}{K_{ij}}\right) + 1\right). \tag{15}$$

For ease of notation, hereafter α_{ij} will be assumed to be equal to 1, $\forall i, j$. Nevertheless, all the following proofs still hold for arbitrary values of $\alpha_{ij} > 0$.

Proposition 1. *Given an initial position $x_0 \in \mathcal{D}^{\epsilon}_{G,\delta}$, for some $\epsilon > 0$, if the system is driven by the control law in Eq.* (4)*, with the edge-weights defined in Eq.* (15)*, the total tension energy of graph G defined in Eq.* (14) *does not increase.*

Proof. From Eqs. (4), (13), (15), the control law of the system can be rewritten as follows:

$$\dot{x}_i = -\sum_{j\in\mathcal{N}_i}\frac{\partial V_{ij}(\delta, x)}{\partial x_i} = -\frac{\partial V}{\partial x_i}. \tag{16}$$

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

Assume that, at time $\tau, x(\tau) \in \mathcal{D}^{\epsilon'}_{G,\delta}$ for some $\epsilon' > 0$. The total tension energy of the graph $V(\delta, x)$, defined in Eq. (14), is a positive function, and is zero only in the desired configuration, i.e. $\|l_{ij}\| = D_{ij}$ $\forall l_{ij} \in E$. The time derivative of the total tension energy function is defined as follows:

$$\dot{V}(\delta, x(\tau)) = \nabla_x V(\delta, x(\tau))^T \cdot \dot{x}(\tau) = -\sum_{i=1}^{N} \dot{x}_i^T \dot{x}_i. \tag{17}$$

Thus, for any $x(\tau) \in \mathcal{D}^{\epsilon'}_{G,\delta}$, $\dot{V}(\delta, x(\tau))$ is non-positive, which proves the statement. □

The total tension energy function $V(\delta, x)$ has been defined in Eq. (14) as the sum of non-negative functions $V_{ij}(\delta, x)$ described in Eq. (10). Since these functions are clearly equal to zero only when the agents are in the desired configuration, it is possible to conclude that the desired formation is a global minimum for the total tension energy $V(\delta, x)$. However, it is possible for the system to evolve to some local minima of the total tension energy. In order to avoid local minima, the Virtual Relabeling algorithm, that will be described in Section 5, may be exploited.

In order to ensure that the presented control algorithm avoids collisions between agents achieving formation, the following proposition is provided.

Proposition 2. *Given an initial position $x_0 \in \mathcal{D}^{\epsilon}_{G,\delta}$, for some $\epsilon > 0$, under the control law in Eq.* (4)*, with the edge-weights defined in Eq.* (15)*, collisions among the agents are always avoided.*

Proof. Let δ be the safety distance for the agents, i.e. if the distance between each couple of agents is greater than δ, collisions are avoided. To prove the statement, it will be shown that as $V(\delta, x(\tau))$ decreases (or at least does not increase), no edge-length will approach δ. To this purpose, define

$$\hat{V}_\epsilon = \max_{x \in \mathcal{D}^{\epsilon}_{G,\delta}} V(\delta, x). \tag{18}$$

Since the function $V(\delta, x)$ is bounded inside the set $\mathcal{D}^{\epsilon}_{G,\delta}$, this maximum exists. Let $M_1 \geq 0$ be the number of edges whose length is less than D_{ij}, and let $M_2 \geq 0$ be the number of edges whose length is greater than or equal to D_{ij}. Thus, the maximum $\hat{V}_\epsilon$ is obtained when M_1 edge-lengths are equal to the minimum allowed length (i.e. $\|l_{ij}\| = (\delta + \epsilon)$) and M_2 edge-lengths are equal to the maximum allowed length (i.e. $\|l_{ij}\| = D_M$). Hence:

$$\begin{aligned}\hat{V}_\epsilon = {} & M_1 \left(\coth\left(\frac{(\delta + \epsilon) - \delta}{K^*} \right) + \frac{1}{2}(\delta + \epsilon)^2 - V_*^{\min} \right) \\ & + M_2 \left(\coth\left(\frac{D_M - \delta}{K^*} \right) + \frac{1}{2} D_M^2 - V_*^{\min} \right),\end{aligned} \tag{19}$$

where K^* and $V_*^{\min}$ are the mean values of K_{ij} and $V_{ij}^{\min}$. A bound will now be defined for the minimal edge-length that can generate this value for the total tension energy. Consider this total tension energy as if it were generated from one single edge, whose edge-length is $\hat{l}_\epsilon \leq (\delta + \epsilon)$, while all the other edge-lengths are equal to D_{ij}, i.e. their contribution to the total tension energy is zero. The edge-length $\hat{l}_\epsilon$ is defined such that

$$\hat{V}_\epsilon = \coth\left(\frac{\hat{l}_\epsilon - \delta}{K} \right) - V_\epsilon^{\min}, \tag{20}$$

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

where $V_\epsilon^{\min} > 0$. To prove the statement, it is necessary to prove that $\hat{l}_\epsilon > \delta$. Substituting Eq. (20) into Eq. (19):

$$\coth\left(\frac{\hat{l}_\epsilon - \delta}{K}\right) = V_\epsilon^{\min} + M_1\left(\coth\left(\frac{\epsilon}{K^*}\right) + \frac{1}{2}(\delta+\epsilon)^2 - V_*^{\min}\right) + M_2\left(\coth\left(\frac{D_M - \delta}{K^*}\right) + \frac{1}{2}D_M^2 - V_*^{\min}\right). \tag{21}$$

From the definition of the edge-tension function in Eq. (10), it follows:

$$\begin{aligned}\coth\left(\frac{\epsilon}{K^*}\right) + \frac{1}{2}(\delta+\epsilon)^2 - V_*^{\min} \geq 0,\\ \coth\left(\frac{D_M - \delta}{K^*}\right) + \frac{1}{2}D_M^2 - V_*^{\min} \geq 0.\end{aligned} \tag{22}$$

Since $M_1 \geq 0$, $M_2 \geq 0$, and $V_\epsilon^{\min} > 0$, it is possible to conclude that

$$\coth\left(\frac{\hat{l}_\epsilon - \delta}{K}\right) \geq 0. \tag{23}$$

Hence, it is possible to conclude that $(\hat{l}_\epsilon - \delta) > 0$, which implies $\hat{l}_\epsilon > \delta$ and, since $\hat{l}_\epsilon$ is bounded by δ, as V decreases (or at least does not increase), no edge-length tends to δ. □

It is important to stress out that Proposition 2 holds for arbitrarily values of $D_M > \delta > 0$. Hence, the value of D_M may be chosen bigger than the radius of the environment where the agents move. To deal with limited communication ranges, this control strategy can be extended, modifying the edge-weight function in Eq. (15) as described in ref. [16].

4. Obstacle Avoidance

One of the main issues that arises when robots have to be coordinated in an unstructured or unknown environment is that they have to take into account the presence of obstacles. The problem of obstacle avoidance has been widely investigated in the literature and typical solutions rely on the possibility of avoiding collisions with environmental obstacles of unknown position by exploiting many different approaches, such as the bug algorithm,[24,25] artificial potential fields,[26] or vector field histogram.[27,28] Even though such approaches have demonstrated their reliability, the purpose of the paper is to define an obstacle avoidance algorithm that can be included in the consensus-based framework presented so far. For this purpose, consider, without loss of generality, the case where a robot detects an obstacle at a distance $d_o \leq d_{\text{sens}}$, where d_{sens} represents the effective range of on-board sensors able to detect obstacles all around the robot. In that case, as described in Olfati-Saber,[18] a virtual agent is projected on the obstacle by the robot that detects it (see Fig. 2). While in Olfati-Saber[18] artificial potential fields are used for collision avoidance purposes, in this paper the virtual agents (i.e. the obstacles) are included into the previously described graph-based algorithm. In particular, the Laplacian-based algorithm used to achieve formation can be extended simply by introducing a new virtual edge to the connectivity graph to represent the link between the real robot and the virtual agent.

For the sake of clarity, suppose that the number of virtual agents detected by the robots of the swarm is N_o. The new graph $G' = (V', E')$ can be considered an extended version of the original one, thus all the sets used to describe graph G must be redefined in order to take into account the virtual edges. The vertex set and the edge set can be redefined as $V' := V \bigcup V_o$ and $E' := E \bigcup E_o$ respectively, where:

- $V_o = \{v_i, i = 1 \dots N_o\}$ is the virtual vertex set,
- $E_o \subseteq V_o \times V_o = \{(v_i, v_k) \in E_o \Leftrightarrow v_i \in V \wedge v_k \in V_o\}$ is the *virtual edge set*, i.e. is the set of the edges connecting a real agent with a virtual one.

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

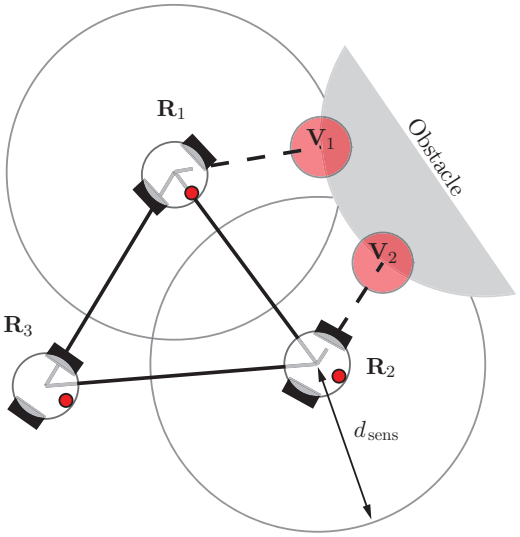


Fig. 2. Three robots moving in formation: two of them (R_1, R_2) detect an obstacle and project on it the corresponding virtual agents (V_1, V_2). Virtual edges are depicted as dashed lines.

As a consequence, also the incidence and weight matrices are modified to include virtual agents. More specifically:

$$\mathcal{I}' = \begin{bmatrix} \mathcal{I} & \mathcal{I}_{o,r} \\ \mathbf{0}_{N_o \times M} & \mathcal{I}_{o,v} \end{bmatrix}, \quad \mathcal{W}' = \begin{bmatrix} \mathcal{W} & 0 \\ 0 & \mathcal{W}_o \end{bmatrix}, \tag{24}$$

where the submatrix $\mathbf{0}_{N_o \times M}$ is a zero matrix with $N_o \times M$ elements and the submatrices $\mathcal{I}_{o,r}$, $\mathcal{I}_{o,v}$ are matrices of $N \times N_o$ and $N_o \times N_o$ elements respectively that summarize the connections between real robots and virtual agents. The elements of $\mathcal{W}$ and $\mathcal{W}_o$ are defined as in Eq. (15), but using two different values of δ such that the distance between robots and obstacles never goes below a predefined *safety* value. In analogy with the definition of the Laplacian matrix reported in Eq. (6), a new Laplacian matrix can be calculated to include the contribution of obstacles in the control law. It is worth noting that, as shown in Fig. 2, the number of virtual agents can be different with respect to the number of detected obstacles. In fact, as long as robots are moving in an unknown environment, they cannot distinguish an obstacle from another one. This means that different robots detecting the same object will project on it different virtual agents.

Assumption 1. *The distance between each couple of obstacles is supposed to be greater then the size of a robot.*

This assumption is necessary to ensure that the obstacles can be overcome by the robots. Otherwise, the swarm can get stuck due to narrow gates. Path planning, which can solve such problems, is beyond the focus of this paper.

Without loss of generality, consider the case where, while N agents are moving in the environment, the pth one senses an obstacle. In this case, the dynamics of the agents that do not sense the obstacle are not influenced directly by its presence. On the contrary, the pth agent defines a virtual agent, whose position corresponds to the position of the obstacle. The dynamics of the virtual agent can be generically described as

$$\dot{x}_v = f_o\left(x_v, t\right). \tag{25}$$

However, as the virtual agent slides on the surface of an obstacle whose shape is not known, it is not possible to define the functions $f_o(\cdot)$ in advance. Furthermore, if the obstacle is not static (e.g. the obstacle is a noncooperative vehicle), its law of motion is supposed to be unknown. As a consequence, the only way to ensure that collisions with obstacles are avoided is to exploit the following proposition.

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

Proposition 3. *The edge-weight function introduced in Eq.* (15) *ensures the avoidance of collisions with obstacles.*

Proof. To include the virtual agent defined once an obstacle is sensed, the total tension energy of the graph G defined in Eq. (14) can be modified as follows:

$$V(\delta, \delta_o, x) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} V_{ij}(\delta, x) + V_{pv}(\delta_o, x), \tag{26}$$

where $V_{pv}(\delta_o, x)$ is the edge-tension function related to the edge between the pth real agent and the vth virtual agent, and δ_o is the safety distance required between robots and obstacles. Namely:

$$V_{pv}(\delta_o, x) = \begin{cases} \alpha_{pv} \left(\coth\left(\dfrac{\|l_{pv}\| - \delta_o}{K_v} \right) + \dfrac{1}{2} \|l_{pv}\|^2 - V_{pv}^{\min} \right) & \text{if } (v_p, v_v) \in E \\ 0 & \text{otherwise} \end{cases},$$

where l_{pv} is the edge vector between the pth real agent and the vth virtual agent, i.e. $l_{pv} = x_p - x_v$, and $\alpha_{pv} > 0$ is a constant value that can be used to modulate the intensity of the interactions between real and virtual agents.

Thus, to prove the avoidance of collisions, Proposition 2 can be applied with this modified total tension energy function. □

Using the same approach, obstacle avoidance can be ensured even in the presence of more than one obstacle sensed by more than one agent (i.e. many virtual agents are added to the graph). Similar to what has been described in Section 3, the choice of the value of the constant $K_v > 0$ is related to the length of the edge between the real and virtual agents for which the edge-weight function is equal to zero. A good choice is to set this distance to the value of the sensing range. In other words, when an agent senses an obstacle and defines a virtual agent, the corresponding edge-weight is always negative, thus always introducing a repulsive action.

It is worth noting that, in the presence of virtual agents, since their movement is not influenced by the position of the real robots, the graph becomes directed. This causes that Proposition 1 does not hold when virtual agents are added to the graph and, as a consequence, this implies that the shape of the formation is not preserved in the presence of obstacles. However, the multi-robot system is not supposed to embed virtual agents inside the formation: virtual agents are introduced only for collision avoidance purposes. Hence, robots will overcome the obstacles without maintaining the shape of the formation, i.e. by performing split-and-rejoin maneuvers, or by reducing the inter-robot distances. Examples of these maneuvers will be shown in the simulations described in Section 6.

5. Local Minima Avoidance

The following section describes an algorithm to let the robots autonomously escape from local minima of the total tension energy function $V(\delta, x)$, introduced in Eq. (14).

5.1. Virtual relabeling

As demonstrated in Proposition 1, the desired formation configuration is the global minimum of the total tension energy function $V(\delta, x)$ introduced in Eq. (6). However, it is possible for the system to evolve to some local minimum configuration. To make the robots escape from local minima, the *virtual relabeling* algorithm may be exploited.

As long as the communication graph is connected, all the robots can calculate the position of the centroid of the group. Then, by computing their own position with respect to the centroid of the group, they can find an agreement on the position they should occupy in the final formation. In order to do this, each robot needs to acquire information from all the other robots of the group. For this purpose, the *data broadcasting* algorithm, which will be introduced in Section 5.2, may be exploited.

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

Table I. Pseudo code for virtual relabeling.

```
01      ind := [1 ... N];
02      while 1 do
03      new_ind := ind;
04      [Xc, Yc] := CalculateCoM();
05      v1 := GetClockwise([Xc, Yc]);
06      v2 := CreateSorted(v1);
07      if v2(i) != ind(i)
08      new_ind(i) := v2(i);
09      end if
10      ind := new_ind;
11      AchieveFormation(ind);
12      end while
```

Therefore, the virtual relabeling algorithm, defined as in Table I, may implemented on each robot. More specifically:

- `[Xc, Yc] := CalculateCoM();` calculates the centroid of the group exploiting both directly acquired and broadcast data.
- `v1 := GetClockwise([Xc, Yc]);` creates a vector where the indices of all the detected robots are saved according to clockwise direction with respect to the centroid.
- `v2 := CreateSorted(v1);` creates a new index vector where all the indices are stored from 1 up to N starting from the previously defined vector `v1`.

To better clarify the relabeling algorithm, in Fig. 3 a typical example of local minimum for a system involving four robots achieving a square formation is depicted. In order to show how the virtual relabeling algorithm can assure the convergence of the swarm of robots to the desired final configuration even in the case of local minima, consider the example reported in Figs. 4(a) and (b). In the first case, the robots reach a local minimum configuration, i.e. according to Proposition 1, the robots reach a configuration where the total tension energy of graph G does not increase, but, due to the wrong order to the robots, is not null. In this case, in fact, the tension energy between each robot and its neighbors results to be null, thus the control law in Eq. (4) is null. Conversely, in the second case, once the virtual relabeling algorithm has been applied, the robots calculate the tension energy corresponding to the desired final formation, i.e. the total tension energy of the group is null if and only if the group reaches the desired formation.

As reported, each robot calculates its own `v1` vector depending on its position with respect to the centroid and teammates. Then, starting from `v1`, vector `v2` is calculated to redefine the actual label that each robot should use to achieve the right configuration. In the depicted example, the robots of the group reach an agreement on `v2` and each of them is able to detect that robots $\mathbf{R_3}$ and $\mathbf{R_4}$ are in the wrong position with respect to their teammates.

5.2. Data broadcasting

One of the main problems related with robots with a limited communication range is that they cannot always acquire information about the whole swarm, that is, some robots can not see each other. Hence, the virtual relabeling algorithm introduced in Section 5.1 cannot be implemented. However, as long as the communication graph is connected, this problem may be overcome by introducing data broadcasting between teammates, i.e. by allowing each robot to transmit information about the relative position of connected teammates with respect to itself, thus transforming *de facto* a connected communication graph into a complete graph. As an example, consider the three robots depicted in Fig. 5. The corresponding Neighbors subsets are defined as

$$\mathcal{N}_i = \{\mathbf{R}_j\} \quad \mathcal{N}_j = \{\mathbf{R}_i,\ \mathbf{R}_k\} \quad \mathcal{N}_k = \{\mathbf{R}_j,\ \mathbf{R}_h\} \quad \mathcal{N}_h = \{\mathbf{R}_k\}\,.$$

Generally speaking, given a couple of communicating robots, it is possible to define $\Delta^x_{ij} = (x_i - x_j)$ and $\Delta^y_{ij} = (y_i - y_j)$ as the relative distance between robots i and j on the x-axis and y-axis respectively. Thus, the ith robot can estimate the relative position of the hth robot by

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

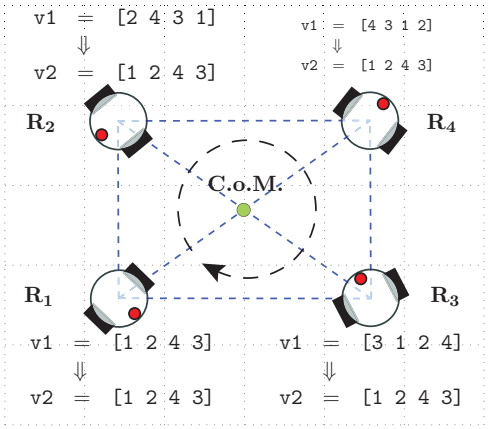
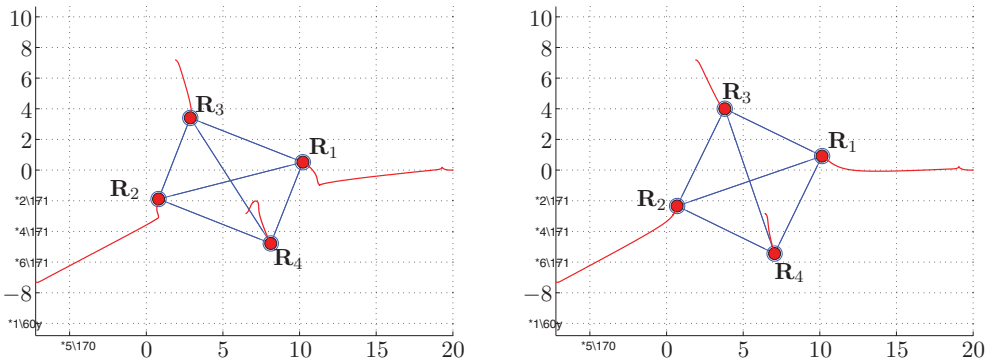


Fig. 3. Example of four robots stuck in a local minimum configuration while achieving a square formation. The indices vectors `v1` and `v2` are defined by the relabeling algorithm.



(a) Simulation run without performing virtual relabeling.

(b) Simulation run performing virtual relabeling.

Fig. 4. Simulation run enlightening different behaviors of a group of robots achieving square formation in case the virtual relabeling is (a) disabled, or (b) enabled.

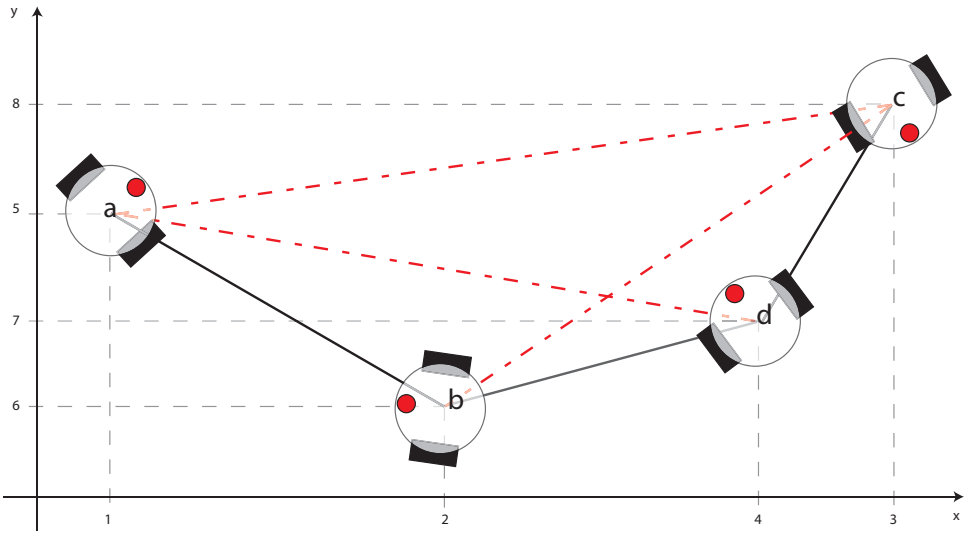


Fig. 5. Example of data broadcasting. Solid lines represent the existing communication links, while dash-dotted lines represent the broadcast links. As an example, robot $\mathbf{R_j}$ broadcasts the values $[(x_k - x_j),\ (y_k - y_j)]^T$, and $[(x_h - x_j), (y_h - y_j)]^T$ to robot $\mathbf{R_i}$, thus allowing it to localize teammates that are not directly seen.

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

exploiting the following equations:

$$(x_h - x_i) = (x_j - x_i) + \Delta_{hj}^x, \quad (y_h - y_i) = (y_j - y_i) + \Delta_{hj}^y.$$

In order to reduce the measurement errors, the data broadcast by each robot is grouped into a data packet containing the sender ID and the relative position of their neighbors with respect to itself. Each of the transmitted data are associated with a *hop count* c_{ij} ($\forall i = 1 \dots N$, $j \in \mathcal{N}_i$) that is incremented each time a robot broadcasts position data that are not directly measured, thus allowing the receiver to chose the data with the lowest *hop count*. The general scheme of the string transmitted by the generic jth robot of a swarm is the $4\|\mathcal{N}_j\| + 2$ elements vector defined as:

Vector element:	**j**	$\|\mathcal{N}_j\|$	i	$x_i - x_j$	$y_i - y_j$	c_{ij}	
Vector index:	0	1	3	4	5	6	,

where $\|\mathcal{N}_j\|$ is the cardinality of the neighbors subset and terms 3–6 are repeated $\forall i \in \mathcal{N}_j$.

As an example, by considering the system depicted in Fig. 5, the data transmitted by $\mathbf{R}_j$ are stored in the following vector:

j	3	i	Δ_{ij}^x	Δ_{ij}^y	1	k	Δ_{kj}^x	Δ_{kj}^y	1	h	Δ_{hj}^x	Δ_{hj}^y	2

.

It is worth noting that the data broadcasting algorithm may be exploited to improve the obstacle avoidance ability of the system as well. In fact, each robot may also broadcast the position of the virtual agents defined when an obstacle is detected, as described in Section 4.

The data broadcasting algorithm requires the communication graph to be connected. In order to ensure connectivity of the communication graph, several strategies may be exploited (see e.g refs. [16, 29–33], and references therein).

6. Simulations and Experiments

To validate the control strategy presented so far, several simulations and experiments have been implemented on differential-drive robots, described by the following kinematic model

$$\begin{cases} \dot{x}_i = u_i \cos(\phi_i) \\ \dot{y}_i = u_i \sin(\phi_i) \\ \dot{\phi}_i = \omega_i \end{cases}.$$

To deal with the fact that this model represents a non-holonomic system, the feedback linearization technique presented in ref. [34] has been considered. Moreover, to make the formation move in a desired direction, a common offset has been added to the control law in Eq. (4) to describe the desired speed of the barycenter of the formation. A video-clip that describes both simulations and experiments can be viewed on the web.[1]

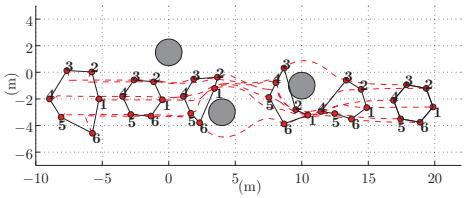
6.1. Simulations

Simulations are performed using Matlab/Simulink. As an example, six robots are simulated while achieving a formation with 1.5-m radius, moving in an environment where three round obstacles are placed on their trajectory. To emphasize the different behaviors that come out by changing the value of α_{ij}, two different simulations are reported in Figs. 6(a) and (b): In the first one, where the inter-robot action is modulated by $\alpha_{ij} = 1$, robots manifest a flexible behavior when obstacles are encountered. In the second one, where $\alpha_{ij} = 10$, the formation is too rigid to be able to split in order to overcome the obstacles, thus the formation preserves its shape while *sliding* over them.

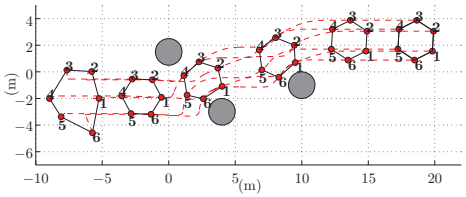
Figure 6(c) shows the mean square error (MSE) between the actual and desired distances for each pair of robots. It is worth noting that, for time 2.5 s $\leq t \leq$ 6 s, i.e. when obstacles are encountered, the MSE value is high if $\alpha_{ij} = 1$ and is approximatively 0 when $\alpha_{ij} = 10$.

[1] The video-clip containing simulations and experiments can be freely viewed and downloaded at http://www.arscontrol.unimore.it/robotica 2013

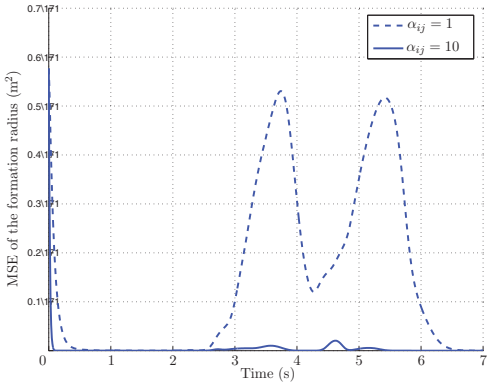
https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press



(a) Soft formation ($\alpha_{ij} = 1$, $\alpha_{pv} = 1$).

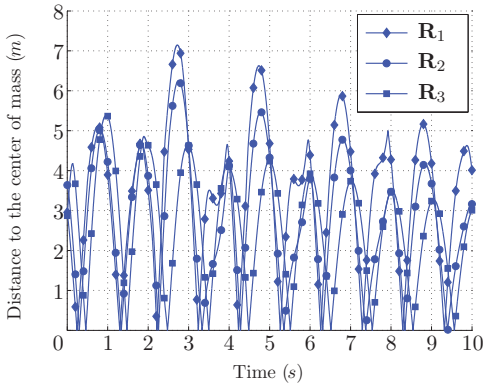


(b) Rigid formation ($\alpha_{ij} = 10$, $\alpha_{pv} = 1$).

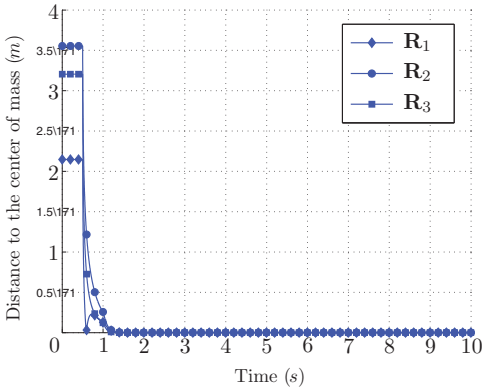


(c) Mean square error (MSE) of the formation radius r with different values of α_{ij}.

Fig. 6. Simulation with six robots engaged in a formation task while moving in an unknown environment with three obstacles (in gray) with different values of α_{ij}.



(a) Formation control algorithm introduced in ref. [8] exploiting artificial potential for collision avoidance.



(b) Formation control algorithm introduced in this paper.

Fig. 7. Error between the actual and the desired distances between each robot and the centroid of the formation exploiting potentials introduced in ref. [8] and the consensus-based algorithm introduced in this paper, considering a communication delay of 0.5 s.

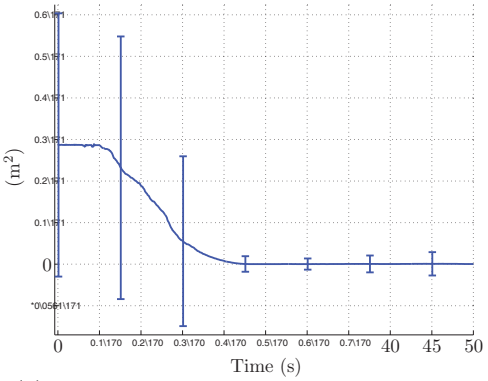
6.2. Communication delay

To better enlighten the behavior of the system in the presence of communication delay, some other simulations involving only three robots achieving regular formation have been performed.

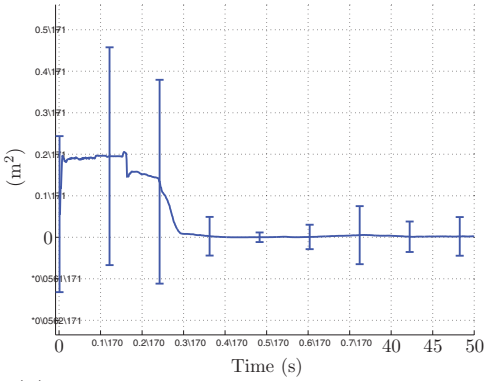
More specifically, simulations show three robots moving in an obstacle-free environment. To compare the algorithm with a potential field-based algorithm, the control law introduced in ref. [8] has been implemented first, adding an artificial potential field for collision avoidance, as described in ref. [9]. Simulations show that, as proven in ref. [15], artificial potential fields are not robust with respect to communication delays. In fact, Fig. 6.1 shows the error between the actual and desired distances between each robot and the centroid of the formation, with a communication delay of 0.5 s. As expected, the system does not converge to the desired formation.

Conversely, simulations show that the control law introduced in this paper is robust with respect to communication delays, as expected for consensus-based control laws.[9] Figure 7(b) shows the error between the actual and desired distances between each robot and the centroid of the formation, with a communication delay of 0.5 s. As expected, the error quickly converges to zero.

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press



(a) Experimental data in a real arena without obstacles.



(b) Experimental data in a real arena with obstacles.

Fig. 8. Average and standard deviation of MSE between the actual and desired distances for each pair of robots, depending on time, in a real arena (a) without obstacles, and (b) with obstacles.

6.3. Experimental results

Experiments are performed within the software platform described in ref. [35]. A group of four E-Puck robots[36] are moving in a 2.0 $\times$ 1.5 m arena. The position of each robot is monitored with a webcam positioned over the arena, and the robots are tracked using colored markers. Hence, position data are collected in a centralized manner. However, the algorithm has been implemented in a decentralized way, simulating the presence of limited range communication. More specifically, the communication range was set to $R = 0.3$ m: each robot was then allowed to exchange data only with its neighbors, that is, robots whose distance was less than R.

The size and position of the obstacles was known in advance to the central computation unit. However, the presence of exteroceptive sensors for obstacle detection was emulated: the position of each obstacle was available to each robot only when the distance between an obstacle and the robot itself was less than the sensing range, $R = 0.3$ m.

Two different experimental setups have been used to test the algorithm. In the first setup, four robots starting from random positions converge to the desired square formation while avoiding collisions, and after 20 s they start moving along the x-axis with a constant speed.

In the second setup, an obstacle is placed in the middle of the arena in a position unknown to the system, thus robots have to overcome it while moving in formation along the x-axis. This setup has been used also to test different behaviors of the system in case of $\alpha_{ij} = 1$ and 10.

For each setup, data have been collected over 10 experimental runs. The average MSE between the actual and desired distances for each pair of robots are represented in Figs. 8(a) and (b) respectively. In particular, in the second setup, it can be seen that at time $\approx$35 s the formation detects the obstacle, thus the variance increases.

7. Conclusions

This paper described a consensus-based algorithm for formation control of groups of mobile robots. By means of the definition of appropriate edge-weight functions, formation control is achieved, as well as inter-robot collision avoidance. The convergence of the system to the desired configuration has been analytically proven, exploiting tools from the graph theory. Moreover, by defining virtual agents projected on the detected unknown objects, the same framework has been exploited to perform obstacle avoidance. The proposed control algorithm has been validated by means of extensive Matlab/Simulink simulations and experiments conducted on real robots moving in an unknown environment.

Future works will aim at explicitly including connectivity maintenance strategies in order to relax requirement on the connectedness of communication graph.

References

1. T. Balch, F. Dellaert, A. Feldman, A. Guillory, C. L. Isbell, Z. Khan, S. C. Pratt, A. N. Stein and H. Wilde, "How multirobot systems research will accelerate our understanding of social animal behavior," *Proc. IEEE* **94**(7), 1445–1463 (Jul. 2006).

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

2. Z. I. Reznikova and B. Ya. Ryabko, "Using the Ideas of the Information Theory to Study Communication Systems of Social Animals," *Proceedings of the IEEE International Symposium on Information Theory, 1995* (Sep. 1995) p. 78.
3. C. W. Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model," *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*, Vol. 21 (ACM, New York, NY, Jul. 1987) pp. 25–34.
4. Y. Liu and K. M. Passino, "Stable social foraging swarm in a noisy environment," *IEEE Trans. Autom. Control* **48**(4), 692–697 (2004).
5. M. Lindhé, P. Ögren and K. H. Johansson, "Flocking with Obstacle Avoidance: A New Distributed Coordination Algorithm Based on Voronoi Partitions," *Proceedings of the IEEE International Conference on Robotics and Automation* (2005) pp. 1785–1790.
6. A. Rosales, G. Scaglia, V. Mut and F. di Sciascio, "Formation control and trajectory tracking of mobile robotic systems – a linear algebra approach," *Robotica* **29**(3), 335–349 (2011).
7. N. Leonard and E. Fiorelli, "Virtual Leaders, Artificial Potentials and Coordinated Control of Groups," *Proceedings of the IEEE Conference on Decision and Control* (2001) pp. 2968–2973.
8. J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control* **49**(9), 1465–1476 (2004).
9. W. Ren, R. Beard and E. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.* **27**(2), 71–82 (2007).
10. M. A. Hsieh, V. Kumar and L. Chaimowicz, "Decentralized controllers for shape generation with robotic swarms," *Robotica* **26**(5), 691–701 (2008).
11. L. Sabattini, C. Secchi and C. Fantuzzi, "Potential-Based Control Strategy for Arbitrary Shape Formations of Mobile Robots," *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems* (2009) pp. 3762–3767.
12. L. Sabattini, C. Secchi and C. Fantuzzi, "Arbitrarily shaped formations of mobile robots: Artificial potential fields and coordinate transformation," *Auton. Robots* (Springer) **30**(4), 385–397 (May 2011).
13. T. Balch and M. Hybinette, "Social Potentials for Scalable Multi–Robot Formations," *Proceedings of the IEEE International Conference on Robotics and Automation* (2000) pp. 73–80.
14. R. Bachmayer and N. E. Leonard, "Vehicle Networks for Gradient Descent in a Sampled Environment," *Proceedings of the IEEE International Conference on Decision and Control* (2002) pp. 112–117.
15. C. Secchi and C. Fantuzzi, "Formation Control Over Delayed Communication Networks," *Proceedings of the IEEE International Conference on Robotics and Automation* (2008) pp. 563–568.
16. M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.* **23**(4), 693–703 (2007).
17. D. V. Dimarogonas and K. H. Johansson, "Stability analysis for multi-agent systems using the incidence matrix: Quantized communication and formation control," *Automatica* **46**(4), 695–700 (2010).
18. R. Olfati–Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control* **51**, 401–420 (2006).
19. T. H. Summers, C. Yu and B. D. O. Anderson, "Robustness to Agent Loss in Vehicle Formations and Sensor Networks," *Proceedings of the IEEE Conference on Decision and Control* (2008) pp. 1193–1199.
20. R. Falconi, S. Gowal and A. Martinoli, "Graph-Based Distributed Control of Non-Holonomic Vehicles Endowed with Local Positioning Information Engaged in Escorting Missions," *IEEE Conference on Robotics and Automation (ICRA 2010)* (2010) pp. 3207–3214.
21. R. Falconi, L. Sabattini, C. Secchi, C. Fantuzzi and C. Melchiorri, "A Graph-Based Collision-Free Distributed Formation Control Strategy," *Proceedings of the IFAC World Congress* (2011) pp. 6011–6016.
22. G. Royle and C. Godsil, *Algebraic Graph Theory* (Springer, New York, NY, 2001).
23. R. M. Murray, R. Olfati-Saber and J. A. Fax, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proc. IEEE* **27**(1), 215–233 (Jan. 2007).
24. V. J. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *IEEE Trans. Syst. Man Cybern.* **20**(5), 1058–1069 (Sep./Oct. 1990).
25. I. Kamon, E. Rimon and E. Rivlin, "Tangentbug: A range-sensor-based navigation algorithm," *Int. J. Robot. Res.* **17**(9), 934–953 (Sep. 1998).
26. O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2 (Mar. 1985) pp. 500–505.
27. J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.* **7**(3), 278–288 (Jun. 1991).
28. I. Ulrich and J. Borenstein, "Vfh+: Reliable Obstacle Avoidance for Fast Mobile Robots. *Proceedings of the IEEE International Conference on Robotics and Automation, 1998*, Vol. 2 (May 1998) pp. 1572–1577.
29. L. Sabattini, N. Chopra and C. Secchi, "On Decentralized Connectivity Maintenance for Mobile Robotic Systems," *Proceedings of the IEEE Conference on Decision and Control* (2011) pp. 988–993.
30. L. Sabattini, N. Chopra and C. Secchi, "Distributed Control of Multi-Robot Systems with Global Connectivity Maintenance," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2011) pp. 2321–2326.
31. L. Sabattini, N. Chopra and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *Int. J. Robot. Res.* (SAGE) **32**(12), 1411–1423 (Oct. 2013).

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press

32. L. Sabattini, C. Secchi, N. Chopra and A. Gasparri, "Distributed control of multi-robot systems with global connectivity maintenance," *IEEE Trans. Robot.* **29**(5), 1326–1332 (Oct. 2013).
33. P. R. Giordano, A. Franchi, C. Secchi and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *Int. J. Robot. Res.* **32**(3), 299–323 (2013).
34. B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics: Modelling, Planning and Control* (Springer, London, 2009).
35. MORE-pucks, MORE-pucks: A software platform for multi-robot experiments. Available at: http://www.arscontrol.unimore.it/morepucks/ (2010). (Accessed December 30, 2013)
36. F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J. C. Zufferey, D. Floreano and A. Martinoli, "The E-puck, a Robot Designed for Education in Engineering," *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions* (2009) pp. 59–65.

https://doi.org/10.1017/S0263574714000368 Published online by Cambridge University Press