Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-13 (2018) 216–222

Priority Task-Based Formation Control and Obstacle Avoidance of Holonomic Agents with Continuous Control Inputs

Miguel A. Trujillo *, **Héctor M. Becerra** **, **David Gómez-Gutiérrez** ***, **Javier Ruiz-León** *, **Antonio Ramírez-Treviño** *

* *CINVESTAV-IPN, Unidad Guadalajara, Zapopan, México (e-mail: {matrujillo, jruiz,art} @gdl.cinvestav.mx)*
** *Centro de Investigación en Matemáticas (CIMAT), Guanajuato, México (e-mail: hector.becerra@cimat.mx).*
*** *Intel Labs, Intel Tecnología de México, Zapopan, México, (e-mail: david.gomez.g@ieee.org).*

Abstract: This work addresses the problem of controlling a network of connected and holonomic agents to achieve a formation while obstacles are avoided. The agents themselves can be obstacles for each other or there may be fixed obstacles in the environment. The proposed control law is computed by the adaptive convex combination of two control laws dealing with two tasks, one devoted to achieve a desired formation and the other focused on obstacle avoidance. The adaptive convex combination is twofold, in one hand it prioritizes the obstacle avoidance task when an agent is in the neighborhood of an obstacle, and in the other hand it maintains a continuous control law. Moreover, the formation is formulated as a consensus problem and a novel finite time control law is used to solve this problem.

Keywords: Consensus, multi-agent system, task-based control

1. INTRODUCTION

The problem of motion generation and control of a group of mobile agents has attracted the attention of several researchers (see for instance Parker (2008), Cao et al. (2013), Chen and Wang (2005)), since its solution allows to face more elaborated tasks that a single agent cannot address. The control of the group of agents requires a communication network allowing the state information of each agent to be shared to a subset of remaining agents and to reach an agreement in certain quantities of interest to carry out the established tasks. The interconnections between agents are represented by a Laplacian matrix and the agents agreement is commonly called consensus (Li and Duan (2014)).

In the analysis of consensus problems, Olfati-Saber and Murray (2004) proposed a linear consensus protocol with asymptotic convergence and demonstrated that the algebraic connectivity of the interaction graph, i.e. the second smallest eigenvalue of the graph Laplacian, determines the convergence rate. Results on finite-time stability like the ones of Bhat and Bernstein (2000) have been exploited in problems of multi-agent systems in order to achieve a high-speed convergence. For instance, finite-time control protocols have been proposed in Wang and Xiao (2010) and Zuo and Tie (2014) to address the consensus problem, where the goal is to drive agents modeled as single integrators to a common state.

As an extension of the results on the consensus problem, Ren and Beard (2004) proposed a scheme of decentralized control for a multi-agent system, where each agent shares its state with its neighbours in order to reach a desired formation. Ren (2007) presented a scheme based on consensus to reach a formation with a virtual agent as a reference. The schemes based on consensus allow to reach a desired formation, however, while the agents are driven to the formation, they may collide with each other. Besides, the environment may have some fixed obstacles. This makes necessary to include an obstacle avoidance strategy to use it when required in order to guarantee that the formation task can be attained.

To deal with the formation problem with obstacle avoidance, Jin and Gans (2017) proposed a control law where the desired formation is achieved through a linear consensus protocol applied to virtual agents. When some agents approach each other within a security distance, the control input of those agents switches to an obstacle avoidance controller. Thus, each agent has a switching scheme between two control laws to achieve a collision-free formation. In the robotics community, a hierarchical task-based control framework has been proposed to address a control problem that involves to solve several tasks simultaneously (Samson et al. (1991); Baerlocher and Boulic (2004)). This scheme has been used in Antonelli et al. (2008) to drive a single agent to a goal while avoiding a

⋆ The second author has been partially supported by Conacyt [grant 220796]. The authors would also like to acknowledge the partial support of Intel Corporation for the development of this work.

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.07.281

fixed obstacle. The same idea has been extended to address formation and obstacle avoidance tasks for multi-agent systems in Antonelli et al. (2008) and Arechavaleta et al. (2017). In those works, the formation control is addressed by solving two tasks: reaching a mean and a variance of the agents positions. Besides, these schemes rely on the global (centralized) sensing of the agents' positions.

In this paper, we propose to tackle the formation control with obstacle avoidance in the framework of hierachical task-based control. The formation control is addressed as a consensus problem of virtual agents such that if the virtual agents reach consensus, then the real agents reach a desired formation. The obstacle avoidance becomes the priority task when the onboard sensing of an agent detects an obstacle closer than a given security distance. We take advantage of previous results of Lee et al. (2012) to provide continuity of the control inputs when both tasks are activated or deactivated. Thus, the contribution of the paper is the formulation of a task function for the formation control and its integration with the obstacle avoidance task in a distributed control scheme, where no global sensing of the agents positions is needed. In addition, a novel nonlinear control to achieve the formation in finite time is introduced.

This work is structured as follows. Section 2 recalls some definitions and results from graph theory. The task-based control scheme is also introduced in this section. In Section 3, the problem of formation control with obstacle avoidance is formulated and solved. Section 4 presents an illustrative example showing the performance of the proposed task-based control. Finally, the main conclusions of this work are presented in Section 5.

2. PRELIMINARIES

2.1 Graph Theory

The following notation and preliminaries on graph theory are taken mainly from Godsil and Royle (2001).

A graph $\mathcal{G}$ consists of a vertex set $\mathcal{V}(\mathcal{G})$ and an edge set $\mathcal{E}(\mathcal{G})$ where an edge is an unordered pair of distinct vertices of $\mathcal{G}$. Writing ij denotes an edge, and $j \sim i$ denotes that the vertex i and vertex j are adjacent or neighbors, i.e., there exists an edge ij. The set of neighbors of vertex i in the graph $\mathcal{G}$ is represented by $\mathcal{N}_i(\mathcal{G}) = \{j : ji \in \mathcal{E}(\mathcal{G})\}$.

A weighted graph includes a weight function $W : \mathcal{E}(\mathcal{G}) \to \mathbb{R}_+$ on its edges. The adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ of a graph with n vertices is a square matrix where a_{ij} corresponds to the weight of the edge ij, when i is not adjacent to j then $a_{ij} = 0$. Through this work it is assumed that $a_{ij} = a_{ji}$, i.e. only *undirected* and *balanced graphs* are considered where $\sum_j a_{ij} = \sum_j a_{ji}$. The Laplacian matrix of $\mathcal{G}$ is $L(\mathcal{G}) = \Delta - A$ where $\Delta = diag(d_1 \cdots, d_n)$ with $d_i = \sum_{j=1}^{n} a_{ij}$.

A path from i to j in a graph is a sequence of distinct vertices starting with i and ending with j such that consecutive vertices are adjacent. If there is a path between any two vertices of the graph $\mathcal{G}$ then $\mathcal{G}$ is said to be connected, otherwise it is said to be disconnected. If the graph $\mathcal{G}$ is connected, then the eigenvalue $\lambda_1(\mathcal{L}) = 0$ has algebraic multiplicity one with eigenvector $\mathbf{1} = [1 \ \cdots \ 1]^T$, i.e. $\ker L(\mathcal{G}) = \{x : x_1 = \ldots = x_n\}$.

Consider a multi-agent system composed of N agents that are connected through a network with single-integrator dynamics given by

$$\dot{x}_i(t) = u_i, \ i \in \{1, \ldots, N\} \tag{1}$$

where $x_i, u_i(t) \in \mathbb{R}$ are the state and the control input of agent i, respectively. The dynamics of the network system (1) can be written in vector form as

$$\dot{x}(t) = u(t) \tag{2}$$

where $x(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$ is the state vector and $u(t) = [u_1(t), u_2(t), \ldots, u_N(t)]^T \in \mathbb{R}^N$ is the control inputs vector of the agents.

If the weighted error of agent i with respect to its neighbors is defined as (Olfati-Saber and Murray (2004)):

$$e_i(t) = \sum_{j\in N_i} a_{ij}(x_j(t) - x_i(t)), \ i \in \{1, \ldots, N\}, \tag{3}$$

then the consensus is achieved when this error es equal to zero for all the agents. The weighted error function (3) can be expressed in matrix form as:

$$e(t) = [e_1(t), e_2(t), \ldots, e_N(t)]^T = -Lx(t). \tag{4}$$

2.2 Multiobjective task-based control

The tasks to be performed by the agents are defined as a function of the current state q of the system. In general, the $i-th$ task function is represented as $x_i(q)$. Since this task must reach an objective x_i^d, then an error function $e(x_i(q), x_i^d)$ can be stated, for instance the simplest one:

$$e_i(q) = x_i(q) - x_i^d \tag{5}$$

where $e_i(q), x_i(q)$ and $x_i^d \in \mathbb{R}^m$. The current task value $x_i(q)$ depends on the current state of the systems $q = [q_1 \ q_2 \ \cdots \ q_N] \in \mathbb{R}^n$.

Taking the derivative of equation (5) with respect to time, the following linear system is obtained

$$\dot{e}_i = J_i(q)\dot{q} \tag{6}$$

where $J_i(q) \in \mathbb{R}^{m\times n}$ is the Jacobian matrix which relates the change in the error with respect to the system velocities. In order to solve for $\dot{q}$, which will be used as the control inputs of the system, the Moore-Penrose pseudoinverse of the Jacobian matrix can be used as in Siciliano (1990)

$$\dot{q} = J_i(q)^+\dot{e}_i \tag{7}$$

where $J_i(q)^+ = J_i(q)^T(J_i(q)J_i(q)^T)^{-1}$. The previous equation generates the minimum norm vector of velocities (Klein and Huang (1983)), i.e. the smallest state variation due to an error variation. Since $\dot{q}$ is the control inputs vector, then the number of control inputs must be larger or equal to the variables to be controlled, i.e. $m \leq n$, which means that the degrees of freedom of the tasks must be less than or equal to the order of the whole system.

In Equation (7), the error dynamics is a design parameter and it must be a stable system, i.e. $\dot{e}_i = f_i(e_i)$ is a stable system. Then Equation (7) becomes:

$$\dot{q} = J_i(q)^+ f_i(x_i(q) - x_i^d). \tag{8}$$

The previous equation indicates the evolution of q ensuring that the $i-th$ task converges to the desired value x_i^d. For instance, consider a case with $n = 2$. Establishing a desired dynamics for (6) as

$$\dot{e}_i = -\lambda e_i \tag{9}$$

where $\lambda > 0$ in order to achieve exponential stability of the task function. According to (9) and (7), the control inputs are given by

$$\dot{q} = -\lambda J_i(q)^+ e_i(q). \tag{10}$$

2.3 Null space based formulation to include several tasks in the control scheme

In real systems there can exist the need of executing several tasks simultaneously and, depending on the condition of the system, some of them must be considered while others should be discarded, particularly if they give contradictory solutions. In multi-task systems a trade-off between all the task must be established.

A fine solution to this problem was introduced by Samson et al. (1991) and applied to robotic systems (Antonelli et al. (2008, 2009); Lee et al. (2012)). The idea is to assign a fixed priority to each task. The task with the highest priority is named the principal task. Then, the solution of the principal task is always used in the computation of the control input and the solution of other tasks is projected into the null space of the principal task. In this way, the solution of lower priority tasks never interferes with the solution of the principal task.

Following Antonelli et al. (2008), the null space of the principal task is computed as

$$N_i = I - J_i^+(q) J_i(q) \tag{11}$$

where $N_i \in \mathbb{R}^{n\times n}$. For instance, in the case that there exist only two tasks x_1 and x_2, where x_1 has the highest priority, then the total control action is computed as (Antonelli et al. (2008, 2009)):

$$\dot{q} = \dot{q}_1 + N_1 \dot{q}_2 \tag{12}$$

where $\dot{q} \in \mathbb{R}^n$ is the control input to the system, $\dot{q}_1 \in \mathbb{R}^n$ is the computed control input by x_1 and $\dot{q}_2$ is the control input computed by x_2.

In order to avoid that the control solution of the priority task modifies the solution of the control solution given by the secondary task, Baerlocher and Boulic (2004) propose an alternative solution given by

$$\dot{q} = J_1^+ \dot{e}_1 + (J_2 N_1)^+ (\dot{e}_2 - J_2 J_1^+ \dot{e}_1) \tag{13}$$

where the product $J_2 N_1$ constrains the domain of J_2 to the null space of J_1.

When only one task is active, a control input of the form (10) is applied with

$$\dot{q} = -\lambda J_2^+ e_2 \tag{14}$$

and when both tasks are activated, the control input (13) is applied. Therefore, there is an undesired effect of the instantaneous switching between the control laws that yields a discontinuity in the control signals. In order to avoid this, Lee et al. (2012) proposed the following control law

$$\dot{q} = \dot{q}_1' + \dot{q}_{1|2} \tag{15}$$

with

$$\begin{aligned}
\dot{q}_1' &= J_1^+ \dot{e}_1' \\
\dot{q}_{1|2} &= (J_2 N_1)^+ (\dot{e}_2 - J_2 J_1^+ \dot{e}_1') \\
\dot{e}_1' &= h(t)\dot{e}_1 + (1 - h(t)) J_1 J_2^+ \dot{e}_2,
\end{aligned}$$

where $\dot{e}_i$ assigns the desired dynamics (9) and $h(t)$ is a smooth time-dependent function varying continuously from 0 to 1. It can be verified that the control input from (14) is identical to the one in (15) for $h(t) = 0$ (Lee et al. (2012)). Thus, this approach provides a continuous solution for $\dot{q}$.

3. SOLUTION TO THE PROBLEM OF FORMATION WITH OBSTACLE AVOIDANCE

This section solves the following problem.

Definition 3.1. Let $\mathcal{A} = \{A_1, ..., A_N\}$ be a set of $N < \infty$ mobile agents and $\mathbf{F}$ be the required formation of the agents given by a vector of relative distances of each agent with respect to an arbitrary common reference frame, and let us consider that there also exists a set of obstacles in the environment. The formation with obstacle avoidance problem (FOAP) consists in finding control inputs yielding a trajectory for each agent such that the formation is reached and the trajectories avoid the obstacles.

The following assumptions are considered:

- Each agent has omnidirectional sensing capability and focuses only on the nearest obstacle, i.e. it processes one obstacle at a time,
- the communication links between agents is modeled by a Laplacian matrix,
- the agents, together with the communication links, describe a connected graph (as defined in Section 2).

According to the FOAP definition, two tasks are needed, one to reach the required formation and the other to avoid obstacles. Since the agent must be traveling to reach the formation avoiding any obstacle, then the task devoted to avoid obstacles is the principal task (herein named local task) and the task devoted to reach the formation is a secondary task (herein named global task).

The following notation is used during the solution of FOAP.

Let $A_i \in \mathcal{A}$, then x_1^i and x_2^i are the local and global tasks associated to A_i. Also, q_i, $\dot{q}_i$ are the position and velocity that should be imposed to agent A_i, and q_o^i is the position of the nearest obstacle to agent A_i.

3.1 Local task (Obstacle avoidance)

The agent A_i must avoid obstacles (an obstacle is a fixed object or another agent) and to do that, it must always maintain a security distance (R) to the object. In order to meet this objective, task $x_1^i = q_i$ is defined (i.e. it computes the position of agent A_i). Then the following task function is stated

$$e_{oi} = \|q_i - q_o^i\| - R \in \mathbb{R} \tag{16}$$

where $q_o^i \in \mathbb{R}^n$ is the position of the obstacle. Notice that $\|q_i - q_o^i\|$ is the distance between agent A_i and an obstacle, which can be measured with a sensor onboard the agent and thus, the global position of the obstacle is not required. The error dynamics is:

$$\dot{e}_{oi} = \frac{(q_i - q_o^i)^T}{\|q_i - q_o^i\|}\dot{q}_i \in \mathbb{R}^n \tag{17}$$

$$\dot{e}_{oi} = J_{oi}(q)\dot{q}_i. \tag{18}$$

Establishing a desired dynamics for (18) as (9) and solving for $\dot{q}_i$ the following equation is obtained

$$\dot{q}_i = -\lambda J_{oi}^+(q)e_{oi}. \tag{19}$$

Since it represents only the contribution of the local task to the position of agent A_i (it does not include the control due to formation), then q_i is renamed in equation (19), as follows, to clarify this fact

$$\dot{q}_{oi} = -\lambda J_{oi}^+(q)e_{oi}. \tag{20}$$

3.2 Global task (Agent's Formation)

As mentioned in the introduction, the formation problem is adressed as a consensus problem. Consider a set of N mobile agents connected through a communication network such that they exchange information with each other, each agent dynamics is given by:

$$\dot{q}_i = u_i(t) \quad i \in 1, ..., N \tag{21}$$

where $u_i \in \mathbb{R}^n$ is the control input for the i-th mobile. Hence, the dynamics of the whole set of agents is represented by:

$$\dot{q} = u(t) \tag{22}$$

where $u = [u_1, u_2, .., u_N] \in \mathbb{R}^{Nn}$.

According to Jin and Gans (2017), agent formation is specified as a set of fixed translation vectors $z_i \in \mathbb{R}^n$ with respect to an arbitrary common reference frame, thus the position of the $i-th$ agent is translated by z_i given a new virtual agent with a new variable q_{z_i} described as

$$q_{z_i} = q_i + z_i \tag{23}$$

where $q_{z_i} \in \mathbb{R}^n, i \in 1, ..., N$ and the network of virtual agents has the same Laplacian matrix than the original network. The virtual agent dynamics is given by

$$\dot{q}_{z_i}(t) = u_{z_i}(t). \tag{24}$$

Applying the control input (24), later defined, to the system (21), the virtual agents reach the consensus and the real agents reach the desired formation with respect to the common reference frame.

Then the weighted error of agent i with respect to its neighbors is

$$e_{z_i} = \sum_{j \in N_i} (a_{ij}(q_{z_j} - q_{z_i})) \tag{25}$$

and the consensus error vector is

$$e_z = [e_{z_1}, e_{z_2}, ..., e_{z_N}] = -Lq_z(t). \tag{26}$$

The next control law is proposed to solve the consensus of the virtual agents, i.e. the real agents reach the desired formation

$$u_z = \dot{q}_z = k\lfloor e_z \rceil^{\frac{1}{2}} = k \begin{bmatrix} \lfloor e_{z_1} \rceil^{\frac{1}{2}} \\ \vdots \\ \lfloor e_{z_N} \rceil^{\frac{1}{2}} \end{bmatrix} \tag{27}$$

where $\lfloor \bullet \rceil^{\frac{1}{2}} = |\bullet|^{\frac{1}{2}}\text{sign}(\bullet)$. The next proposition states that the proposed control law achieves the consensus (hence also the formation) in finite time.

Proposition 1. Consider a connected undirected (balanced) graph $\mathcal{G}$ and the vector error (e_z) given in equation (26). Then, there exists a control gain $k \in \mathbb{R}$ such that the following nonlinear control protocol

$$u_{z_i}(t) = \dot{q}_{z_i} = k\lfloor e_{z_i} \rceil^{\frac{1}{2}} \tag{28}$$

achieves finite-time convergence to zero of the error vector in equation (26) and consequently the consensus of the state of system (24), for any initial state $q_z(0)$.

Proof. Consider the error dynamics

$$\dot{e}_z = -kL\Phi(e_z) \tag{29}$$

where $\Phi(e_z)$ is given by

$$\Phi(e_z) = \begin{bmatrix} \lfloor e_{z_1} \rceil^{\frac{1}{2}} \\ \vdots \\ \lfloor e_{z_N} \rceil^{\frac{1}{2}} \end{bmatrix} \tag{30}$$

and let

$$V(e) = \sum_{i=1}^{N} |e_{z_i}| \tag{31}$$

be a candidate Lyapunov function for (29), which is positive definite but not continuously differentiable for all e_z. However, since (31) is Lipschitz continuous, the global stability of (29) is obtained if $\dot{V}$ is negative definite almost everywhere (Bacciotti and Rosier, 2006, p. 207), which will be demonstrated in the sequel. To this aim, let $S(e_z) = [\text{sign}(e_{z_1}) \ \cdots \ \text{sign}(e_{z_N})]^T$ and notice that if $e_{z_i} \neq 0$ then the $i-th$ element of $S(e_z)^T L$ is either zero or it has the same sign as e_{z_i}. Then, $S(e_z)^T L\Phi(e_z) = S(e_z)^T W(t)\Phi(e_z)$ where $W(t) = diag(w_1(t), \dots, w_N(t))$, $w_i(t) \geq 0$. Then

$$\dot{V} = -S(e_z)^T L\Phi(e_z) = -k\sum_{i=1}^{N} w_i(t)|e_{z_i}|^{\frac{1}{2}}. \tag{32}$$

Notice that, with $e_{z_i} \neq 0$, $w_i(t)$ is zero iff $\forall j \in \mathcal{N}_i(\mathcal{G})$, $\text{sign}(e_{z_i}) = \text{sign}(e_{z_j})$. Moreover, since $e_z = -Lq_z$ then along the evolution of the system it holds that $\mathbf{1}_N e_z = \sum e_{z_i} = 0$ and therefore, unless $e_z = 0$ there always exists a $e_{z_i} \neq 0$ with $w_i(t) \neq 0$, i.e. a node with nonzero weighted error such that $\exists j \in \mathcal{N}_i(\mathcal{G})$, $\text{sign}(e_{z_i}) \neq \text{sign}(e_{z_j})$. Thus, the origin of (29) is globally asymptotically stable. Notice that if $e_z = Lq_z = 0$ then $q_{z_1} = \ldots = q_{z_n}$ and consensus is achieved.

Since (29) is globally asymptotically stable and for all $\lambda > 0$, $\Phi(e_z) = \lambda^{-(d+1)}\Phi(\lambda e_z)$ with $d = -\frac{1}{2}$ (i.e. the vector field is homogeneous with negative degree with respect to the standard dilation, see e.g. Bhat and Bernstein (2005)), finite-time stability follows from (Bhat and Bernstein, 2005, Theorem 7.1). Thus, consensus is achieved in finite-time. □

Moreover, the result presented in (Olfati-Saber and Murray (2004)) ensures that the consensus of virtual agents is the average of the initial conditions of the agents, when the Laplacian matrix (L) comes from a strongly connected graph, i.e.:

$$\lim_{t \to \infty} q_{z_i}(t) = \alpha\mathbf{1} = q_z^* \tag{33}$$

with

$$\alpha = \frac{\sum_i \gamma_i q_{z_i}(0)}{\sum_i \gamma_i} \tag{34}$$

where γ_i is the $i-th$ element of the left eigenvector γ of L.

Although one may be tempted to assign $x_2^i = q_{z_i}$ and its task error as Equation (25), this is not possible since the Jacobian matrix appearing in the derivative of (25) is quite difficult to manipulate. Instead of that, this work takes advantage of Equation (33) and it defines the secondary task as $x_2^i = q_{z_i}$ (the position of the virtual agents) and the error of this task as

$$e_{ce} = q_z - (q_z^* \otimes I_N)\mathbf{1} \tag{35}$$

and its derivative is given by

$$\dot{e}_{ce} = J_{ce}(q)\dot{q}_z \tag{36}$$

where $(q_z^* \otimes I_N)\mathbf{1} \in \mathbb{R}^{Nn}$ is a vector of ones scaled to the initial conditions and $J_{ce} = I_{Nn} \in \mathbb{R}^{Nn\times Nn}$ is the Jacobian matrix of the consensus task.

Notice that e_{ce} and $\dot{e}_{ce}$ converge to zero in finite time, since the limit in Equation (33) is reached in finite time. Moreover, the resulting error dynamics will be used as the desired one in the priority task-based scheme, i.e.

$$\dot{e}_{ce}^d = k\lfloor e_z \rceil^{\frac{1}{2}}. \tag{37}$$

3.3 Convex combination of global and local tasks

The proposed combination of global and local tasks is an application of that proposed in Lee et al. (2012). Thus, the control input is given by

$$\dot{q} = \dot{q}_1' + \dot{q}_{1|2} \tag{38}$$

where:

$$\begin{aligned}
\dot{q}_1' &= J_o^+ \dot{e}_1' \\
\dot{q}_{1|2} &= (J_{ce}N_o)^+(\dot{e}_{ce} - J_{ce}J_o^+\dot{e}_1') \\
\dot{e}_1' &= h(t)\dot{e}_o + (1-h(t))J_oJ_{ce}^+\dot{e}_{ce}
\end{aligned}$$

and $0 \le h(t) \le 1$ is a smooth function. Depending on this function, the smoothness of $\dot{q}$ can be achieved. The next proposition states that this control law achieves the formation and obstacle avoidance.

Proposition 2. Consider a strongly connected and not directed graph. For this kind of graphs, the control law (38) *guarantees convergence to zero of the consensus error* (e_{ce}) *and obstacle avoidance error* (e_o) *in spite that both tasks are active during the transition to activate/deactivate the obstacle avoidance task. The terms of (38) are*

$$\begin{aligned}
\dot{e}_o &= -\lambda e_o \\
\dot{e}_{ce} &= k\lfloor e_z \rceil^{\frac{1}{2}} \\
J_c &= I_n \\
(J_{ce}N_o)^+ &= \begin{bmatrix} (J_cN_{o1})^+ & 0 & \cdots & 0 \\ 0 & (J_cN_{o2})^+ & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (J_cN_{oN})^+ \end{bmatrix} \\
J_o^+ &= \begin{bmatrix} J_{o1}^+(q) & 0 & \cdots & 0 \\ 0 & J_{o2}^+(q) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_{oN}^+(q) \end{bmatrix}.
\end{aligned}$$

Proof. Let us propose the vector of errors

$$e' = \begin{bmatrix} e_{ce} \\ e_o \end{bmatrix}, \tag{39}$$

where $e_o = \left[e_{o1}^T, e_{o2}^T, \cdots e_{oN}^T\right]^T \in \mathbb{R}^N$ with e_{oi} defined in (16), so e_o is a vector with the task of obstacle avoidance for each mobile and e_{ce} defined in (35) is the consensus of virtual agents (i.e. the formation of real agents). Then, consider a Lyapunov candidate function

$$V = \frac{1}{2}e'^T e' \tag{40}$$

with time derivative

$$\dot{V} = e'^T \dot{e}'. \tag{41}$$

Expanding the obstacle task error for the N agents we get

$$\dot{e}_o = J_o(q)\dot{q} = \begin{bmatrix} J_{o1}(q) & 0 & \cdots & 0 \\ 0 & J_{o2}(q) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_{oN}(q) \end{bmatrix}\dot{q} \tag{42}$$

where $J_o(q) \in \mathbb{R}^{N\times Nn}$ has the Jacobians $J_{oi}(q) \in \mathbb{R}^n$ of the $i-th$ agent. Let $e_{ce}^d = k\lfloor e_z \rceil^{\frac{1}{2}}$ and $\Delta = \mathrm{diag}\{|e_{z_1}(t)|^{-1/2}, \ldots, |e_{z_N}|^{-1/2}\}$. Since

$$|e_{z_i}(t)|^{1/2}\mathrm{sign}(e_{z_i}(t)) = \frac{|e_{z_i}(t)|}{|e_{z_i}(t)|^{1/2}}\mathrm{sign}(e_{z_i}(t)) = \frac{e_{z_i}(t)}{|e_{z_i}(t)|^{1/2}},$$

we have $\lfloor e_z \rceil^{\frac{1}{2}} = \Delta(e_z)e_z$. Furthermore, according to the properties of the Laplacian matrix we have $e_z = -Le_{ce}$, therefore

$$\dot{e}_{ce} = -\Delta(e_z)Le_{ce}. \tag{43}$$

Now, substituting (36) and (42) in (41), we have

$$\dot{V} = \begin{bmatrix} e_{ce}^T & e_o^T \end{bmatrix} \begin{bmatrix} J_{ce}(q) \\ J_o(q) \end{bmatrix} [\dot{q}]. \tag{44}$$

Using the terms of (38), then

$$\dot{V} = \begin{bmatrix} e_{ce}^T & e_o^T \end{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix} \begin{bmatrix} e_{ce} \\ e_o \end{bmatrix} \tag{45}$$

where:

$$\begin{aligned}
M_1 =& k(1-h)J_oJ_o^+J_{ce}^+L\Delta \\
&- J_{ce}(J_cN_o)^+(k(1-h)J_oJ_o^+J_{ce}^+L\Delta + kL\Delta), \\
M_2 =& -(\lambda hJ_{ce}J_o^+ - \lambda hJ_{ce}(J_cN_o)^+J_{ce}J_o^+), \\
M_3 =& -kJ_oJ_o^+(1-h)J_oJ_{ce}^+L\Delta \\
&+ kJ_{oe}(J_cN_o)^+(I-(1-h)J_{ce}J_o^+J_oJ_{ce}^+)L\Delta, \\
M_4 =& \lambda hJ_oJ_o^+ + \lambda hJ_o(J_cN_o)^+(J_{ce}J_o^+),
\end{aligned}$$

recalling that $J_{oi}(q)J_{oi}^+(q) = I_n \in \mathbb{R}^{n\times n}$, $(L\otimes I)q_z^* = 0$, $J_{oi}N_{oi} = 0$, $N_{oi} = N_{oi}^T$ and the products

$$\begin{aligned}
J_{oi}(J_cN_{oi})^+ &= J_{oi}N_{oi}^T(N_{oi}N_{oi}^T)^{-1} \\
&= J_{oi}N_{oi}(N_{oi}N_{oi}^T)^{-1} \\
&= 0, \\
(J_cN_{oi})^+ &= (J_cN_{oi})^T((J_cN_{oi})(J_cN_{oi})^T)^{-1} \\
&= N_{oi}(N_{oi}N_{oi}^T)^{-1} \\
&= I_n.
\end{aligned} \tag{46}$$

Then the derivative of the Lyapunov function is

$$\dot{V} = -\begin{bmatrix} e_{ce}^T & e_o^T \end{bmatrix} M \begin{bmatrix} e_{ce} \\ e_o \end{bmatrix}, \tag{47}$$

where $M = \begin{bmatrix} kL\Delta & O \\ k(1-h(t))J_oL\Delta & \lambda hI_n \end{bmatrix}$. The eigenvalues of matrix M depend on the constants values k, λ, and on $h(t)$ and matrices L and Δ. With $k > 0$, and according to Section 2, the matrix L represents a connected and balanced graph, it has an eigenvalue $\lambda_1(L) = 0$ with an associated eigenvector $\mathbf{1} = [1 \cdots 1]^T$ such that $L\mathbf{1}^T = 0$, which implies that $e_{ce_1} = \cdots = e_{ce_n}$. Moreover, if the matrix $L\Delta$ is not balanced, but L has a left eigenvector γ^T associated with $\lambda_1(L) = 0$ that satisfies $\gamma^T L = 0$, then $\gamma^T L\Delta = 0$ with $\gamma = [\gamma_1, \ldots, \gamma_n]$ and $\gamma_1 = \cdots = \gamma_n$ then $\gamma^T = e_{ce}^T$, which means that consensus is achieved. Since $\lambda > 0$ and $0 \leq h(t) \leq 1$, then the matrix M is semipositive defined with one of the eigenvalues equal to zero and the others are positive. Thus, $\dot{V} < 0$ and when the consensus is achieved with $e_{ce} = 0$ and obstacle is present with $h > 0$ and the error $e_o = 0$, then $\dot{V} = 0$. □

The previous proposition guarantees that the consensus error converges to zero and consequently the desired formation is achieved in finite-time in spite of the occurrence of an obstacle avoidance task.

4. SIMULATION RESULTS

In this section we illustrate the obtained results, where the objective is to achieve a formation of the agents and at same time avoiding the presented obstacles. For the simulation example, we consider a five-agent system ($N = 5$) in the network with undirected graph (Fig. 1). In the simulation, we set $a_{ij} = 1$ and $k = 0.75$. Consider the initial conditions $q_x(0) = [2\ 3\ 3\ 5\ 6]^T$ for the x axis and $q_y(0) = [-2\ -3\ 7\ 5\ 1]^T$ for the y axis of agents 1 to 5 respectively. The environment has 3 fixed obstacles with different positions, the positions are $q_{o1} = [3.7\ 4]$, $q_{o2} = [2\ 1]$ and $q_{o3} = [4.5\ 1.2]$ for obstacle 1, 2 and 3 respectively, and the security distance for obstacles and agents is $R = 0.5$. The displacement vectors for the virtual agents are $z_i = [1.5\cos((72 * i)^\circ),\ 1.5\sin((72 * i)^\circ)], i \in 1, ..., N$ with respect to the center of formation, for each agent respectively.

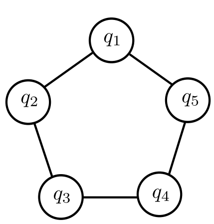


Fig. 1. Connected graph used in the simulations.

Applying the reported results, the numerical simulations for the two tasks of the agents, consensus to achieve a formation and obstacle avoidance, are presented in Figs. 2-5. The Fig. 2 shows the trajectory of virtual agents while the Fig. 3 shows the error between each agent and the agents that are connected, in this case the consensus has a displacement with respect to 0, which implies that the desired formation is achieved. The trajectory of each agent using (38) is shown in Fig. 4, where we can observe that the agents avoid the obstacles (q_o), and when an agent detects some obstacle, it changes the profile of its velocity without discontinuities to avoid the obstacle. The profiles of the velocities of each agent are shown in Fig. 5, note that the profiles of velocities are continuous.

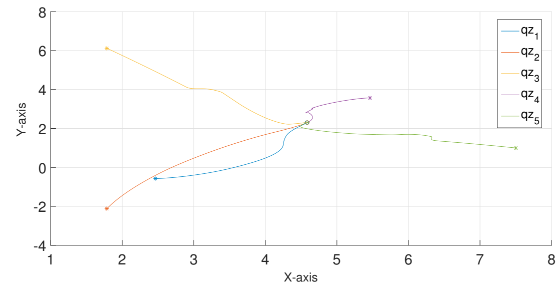
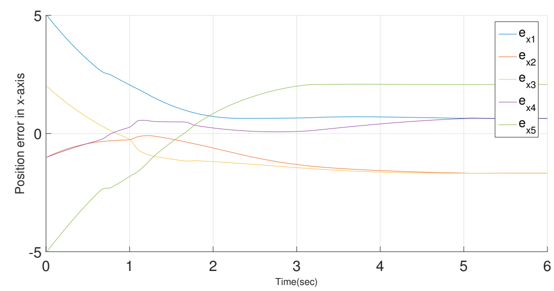
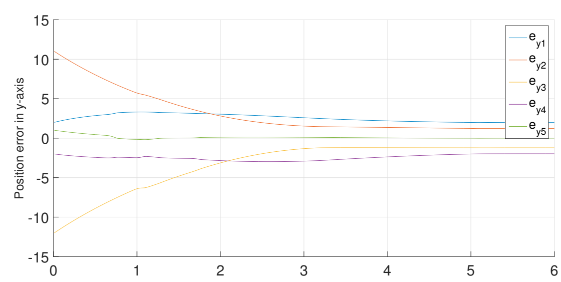


Fig. 2. Trajectories of virtual agents.



(a) Consensus of agents in $x - axis$



(b) Consensus of agents in $y - axis$

Fig. 3. Consensus of agents.

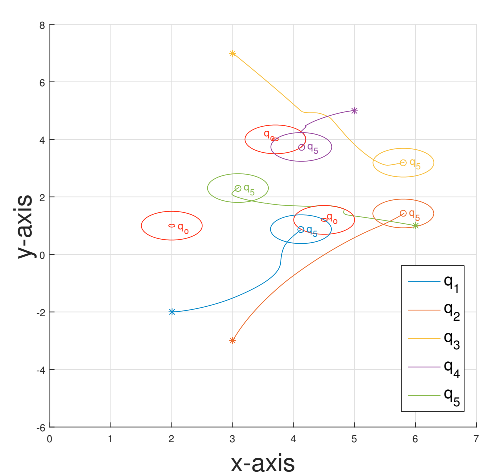


Fig. 4. Trajectories of agents and the reached formation.

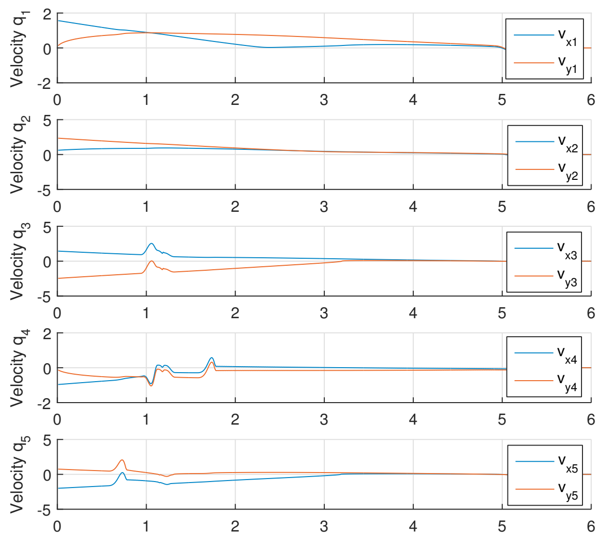


Fig. 5. Control inputs (velocities) of agents.

5. CONCLUSIONS

This paper solved the problem of the formation of N holonomic agents moving on a plane with obstacle avoidance, using the hierarchical task-based scheme for this purpose. In this case two tasks are considered, the high priority one is devoted to avoid obstacles and the low priority one to the agents formation. The formation control has been solved by a novel finite-time control protocol and the formation is achieved independently that some agents have to solve an obstacle avoidance task. We have formulated an adequate task function for the formation problem and it has been combined with the obstacle avoidance task in a hierarchical distributed control approach. The solution interference between both tasks is avoided in this scheme and a continuous time-dependent function $h(t)$ is used to maintain the continuity of the control inputs. The proposed control scheme only needs local (onboard) sensing of the relative agent's positions. The convergence of the proposed solution is proved and its effectiveness is shown in an illustrative example.

REFERENCES

Antonelli, G., Arrichiello, F., and Chiaverini, S. (2009). Experiments of formation control with multirobot systems using the null-space-based behavioral control. *IEEE Transactions on Control Systems Technology*, 17(5), 1173–1182.

Antonelli, G., Arrichiello, F., and Chiaverini, S. (2008). The null-space-based behavioral control for autonomous robotic systems. *Intelligent Service Robotics*, 1(1), 27–39.

Arechavaleta, G., Morales-Daz, A., Prez-Villeda, H.M., and Casteln, M. (2017). Hierarchical task-based control of multirobot systems with terminal attractors. *IEEE Transactions on Control Systems Technology*, 25(1), 334–341.

Bacciotti, A. and Rosier, L. (2006). *Liapunov functions and stability in control theory.* Springer Science & Business Media.

Baerlocher, P. and Boulic, R. (2004). An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *Visual Computer*, 20(6), 402–417.

Bhat, S.P. and Bernstein, D.S. (2000). Finite-time stability of continuous autonomous systems. *SIAM Journal on Control and Optimization*, 38(3), 751–766.

Bhat, S.P. and Bernstein, D.S. (2005). Geometric homogeneity with applications to finite-time stability. *Mathematics of Control, Signals and Systems*, 17(2), 101–127.

Cao, Y., Yu, W., Ren, W., and Chen, G. (2013). An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1), 427–438.

Chen, Y.Q. and Wang, Z. (2005). Formation control: a review and a new consideration. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3181–3186.

Godsil, C. and Royle, G. (2001). *Algebraic Graph Theory*, volume 8 of *Graduate Texts in Mathemathics.* Springer-Verlag New York.

Jin, J. and Gans, N. (2017). Collision-free formation and heading consensus of nonholonomic robots as a pose regulation problem. *Robotics and Autonomous Systems*, 95, 25–36.

Klein, C.A. and Huang, C.H. (1983). Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(2), 245–250.

Lee, J., Mansard, N., and Park, J. (2012). Intermediate Desired Value Approach for Task Transition of Robots in Kinematic Control. *IEEE Transactions on Robotics*, 28(6), 1260–1277.

Li, Z. and Duan, Z. (2014). *Cooperative control of multi-agent systems: a consensus region approach.* CRC Press.

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

Parker, L.E. (2008). *Multiple Mobile Robot Systems*, 921–941. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ren, W. (2007). Multi-vehicle consensus with a time-varying reference state. *Systems and Control Letters*, 56(7), 474 – 483.

Ren, W. and Beard, R. (2004). Decentralized Scheme for Spacecraft Formation Flying via the Virtual Structure Approach. *Journal of Guidance, Control, and Dynamics*, 27(1), 73–82.

Samson, C., Borgne, M.L., and Espiau, B. (1991). *Robot Control: The Task Function Approach*, volume 22 of *Oxford Engineering Science Series.* Oxford University Press, New York, USA, first edition.

Siciliano, B. (1990). Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3(3), 201–212.

Wang, L. and Xiao, F. (2010). Finite-time consensus problems for networks of dynamic agents. *IEEE Transactions on Automatic Control*, 55(4), 950–955.

Zuo, Z. and Tie, L. (2014). A new class of finite-time nonlinear consensus protocols for multi-agent systems. *International Journal of Control*, 87(2), 363–370.