J Control Autom Electr Syst (2015) 26:201–214
DOI 10.1007/s40313-015-0170-x

The Multilayer Control Scheme: A Strategy to Guide *n*-Robots Formations with Obstacle Avoidance

Alexandre Santos Brandão · Vinícius Thiago Lecco Rampinelli · Felipe Nascimento Martins · Mário Sarcinelli-Filho · Ricardo Carelli

Received: 17 May 2014 / Revised: 4 February 2015 / Accepted: 9 February 2015 / Published online: 18 February 2015

Abstract A multilayer scheme is here proposed to implement coordinated control of a group of mobile robots. Specifically, a trajectory tracking controller is proposed to coordinately guide a platoon of robots, with an obstacle deviation subsystem based on virtual forces and mechanical impedance implemented in each robot. Such a controller is firstly designed for groups of three robots and then generalized to larger groups ($n > 3$ robots), by proposing a modular structure based on the concatenation of triangular modules, composing a polygon of n sides (one of the designed controllers is adopted to guide each module). The stability of the closed-loop control system is also proven for each individual control system, based on the theory of Lyapunov, taking into account the saturation of the control signals, adopted to prevent the saturation of the robot actuators. Based on such a proof, a conjecture that the whole control system is stable is presented, which is supported by several simulated and experimental results, some of which are presented, thus validating the proposed control structure.

Keywords Formation control · Mobile robots · Theory of Lyapunov

1 Introduction

A multi-robot system is considered cooperative when it presents a behavior that increases the utility of the agents to accomplish a given task (Cao et al. 1997). Such a cooperative behavior is exemplified in Hess et al. (2009), where a framework is proposed to guide multiple groups of autonomous snowplow robots to remove the snow in airfields. The main motivation to deal with cooperative behavior (groups of robots working cooperatively to accomplish a task) is that a coordinated group of mobile robots can execute several tasks [e.g., surveillance of large areas (Hougen et al. 2000; Stoeter et al. 2002), search and rescue (Jennings et al. 1997), objects pushing (Golkar et al. 2009), and transportation of large objects (Stouten and de Graaf 2004)] in a more efficient way than a single specialized robot (actually, there are tasks simply not accomplishable by a single mobile robot, demanding a coordinated group of robots to perform it). In such context, the term formation control arises, defined as the control of the relative poses of the robots in a platoon, moving as a single structure (Consolini et al. 2007).

Three different techniques have been adopted to control a group of mobile robots to keep a certain geometric formation. They are the leader–follower formation, the use of virtual

The authors thank FAPES (a foundation of the Secretary of Science and Technology of the State of Espirito Santo, Brazil), CAPES (a foundation of the Brazilian Ministry of Education), and SPU (a secretary of the Argentine Ministry of Education), for the support given to this work.

A. S. Brandão (✉)
Department of Electrical Engineering, Universidade Federal de Viçosa, Av. P. H. Rolfs, S/N, DEL, Campus UFV,
Viçosa, MG, Brazil
e-mail: alexandre.brandao@ufv.br

V. T. L. Rampinelli · M. Sarcinelli-Filho
Graduate Program on Electrical Engineering, Federal University of Espírito Santo, Vitória, ES, Brazil
e-mail: mario.sarcinelli@ufes.br

F. N. Martins
Federal Institute of Education, Science and Technology of Espírito Santo (IFES), Serra, ES, Brazil
e-mail: felipe.n.martins@gmail.com

R. Carelli
Institute of Automatics, National University of San Juan,
San Juan, Argentina
e-mail: rcarelli@inaut.unsj.edu.ar

structures, and the behavioral approach (Dong et al. 2006). In the leader–follower formation, a robot referred to as the leader is responsible for guiding all the other robots involved in the formation, so that they reach their desired positions and keep the formation thus composed while moving. In such kind of structure, the leader robot is the most important one, since its failure would prevent the accomplishment of the programmed task (Ogren and Leonard 2003; Shao et al. 2005; Chen and Wang 2005).

Considering a virtual structure, the whole formation is the most important element, since it is dealt with as one rigid body moving in such a way that preserves a geometric figure previously defined (Jia et al. 2006; Gava et al. 2007).

For the behavioral approach, a set of procedures is defined for each robot in the team, such as to seek for the goal, to avoid obstacles, and to establish the formation. The control signals sent to each individual robot in the platoon correspond, in general, to a weighted mean considering the available behaviors (Balch and Arkin 1998; De-Gennaro and Jadbabaie 2006; Liu et al. 2006).

In terms of the control structure, it can be implemented in a centralized or in a decentralized way. In a centralized structure (Cruz and Carelli 2008; Mas and Kitts 2009), the leader of the formation is responsible for getting all the information related to all the robots in the team and for sending the suitable control signals to each one, in order to establish the desired formation. In this case, the centralizing agent runs a single controller that is responsible for guiding all the robots during the navigation. If a decentralized structure is adopted (Chen and Wang 2005; Cruz and McClintock 2007; Brandao et al. 2009), a centralizing agent is not necessary: Each robot in the formation has its own sensors to get the information about its pose, the state of the environment surrounding it and the poses of the other robots (normally related to its system of coordinates), and generates its own control signals to reach the pose necessary to compound the desired formation.

In spite of its intrinsic problems, such as communication between the robots and scalability for a greater number of robots, the centralized formation control has technical advantages when applied to the control of formations whose geometric form is not allowed to change (Antonelli et al. 2008). Some of such advantages are as follows: (1) the robots need to run just direct motor controllers, which reduces the demand for onboard computational power; (2) the whole platoon acts as a single agent, so more complex control algorithms can be implemented; and (3) an optimal solution can be found for the whole platoon, since the controller has information about all of its robots. However, because the centralized controller should receive information from and send commands to all agents in the platoon, a communication scheme is necessary. As an example, in Mas and Kitts (2009), a control approach based on a virtual structure (a centralized control scheme), called Cluster Space Control, is proposed. There, the positioning control is carried out considering the centroid of a triangular robot formation. The framework deals with local variables (the position of each individual robot) and a few parameters related to the formation itself, and relates the correspondent velocities through a Jacobian matrix, which should be rewritten each time a new robot is included in the formation (the system is not scalable).

Following this approach, this work addresses the problem of scalability of the centralized formation control using virtual structures. The main contribution is a generalization of the control approach of Mas and Kitts (2009), which only deals with three robots, to allow controlling any formation containing $n > 3$ robots. Our strategy considers $n-2$ groups of three robots (triangular cells), thus neither demanding increasing the dimension of the Jacobian matrix nor demanding changing its entries. Actually, as this strategy considers only triangular cells, the 3×3 Jacobian matrix is reproduced for each triangular cell, so that $n-2$ identical Jacobian matrices are used. Hence, this proposal does not increase the complexity of the control strategy already designed for a group of three robots, which is only repeated $n - 2$ times. To do that, the multilayer control scheme proposed in Brandão et al. (2009) to control a formation of 3 robots is adapted to control a formation containing $n-2$ triangular cells, or n robots. In such a scheme, similar to the one proposed in Fierro and Das (2002), each layer is responsible for part of the control of the formation. A second contribution of the paper is that the nonlinear controller proposed embeds saturation terms to guarantee that the control signals will be always compatible with the limits of the robot actuators (thus, nonlinearities caused by the saturation of the robot actuators are prevented). A conjecture that the equilibrium of the whole system thus implemented, including the saturation of the control signals, is stable in the sense of Lyapunov is also presented, based on a reasoning centered in the proof of the stability of each triangular module, which is the third contribution of the paper.

In addition, a technique based on the concept of mechanical impedance (Hogan 1985) is embedded in the control structure, to allow avoiding obstacles. To implement such strategy, each individual robot has an obstacle avoidance module associated with it, which means that each robot can independently avoid obstacles while navigating to keep the formation. The idea underlying such system conception is that the task the formation is accomplishing allows the deformation of the robot platoon (surveillance in large areas, for instance). Under this assumption, the obstacle avoidance subsystem implemented allows the robots to individually avoid obstacles in their navigation routes, thus guaranteeing that the entire formation avoids these obstacles.

To discuss such topics, the paper is hereinafter organized as follows: Sect. 2 describes the inverse kinematics of differential drive mobile robots and its generalization for a multirobot system. Notice that differential drive is not the only

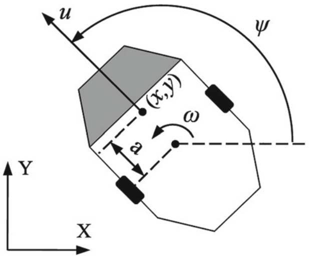


Fig. 1 The differential drive mobile robot

kinematic configuration accepted by the control structure here proposed. Actually, any vehicle moving in a horizontal plane (flat surface), controlled through linear and angular velocities (as high-level control signals), is admitted. In the simulations and experiments here reported, however, just differential drive robots have been used. By its turn, Sect. 3 presents the multilayer control scheme. In the sequel, Sect. 4 presents the proposed control law, the stability analysis of the closed-loop system using such controller and the generalization of the control structure for a formation of $n > 3$ robots. A brief comment on the off-line planning layer is given in Sect. 5, whereas Sect. 6 describes the decentralized strategy proposed to allow the formation to avoid obstacles while tracking a trajectory. Closing the manuscript, Sect. 7 presents some simulated and experimental results and Sect. 8 highlights the main conclusions of the work.

2 Robot Kinematic Model

Differential drive mobile robots, as depicted in Fig. 1, are those considered here, for their good mobility and simple configuration. This way, the kinematic model of the i-th robot in the formation is described as

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\psi}_i \end{bmatrix} = \begin{bmatrix} \cos\psi_i & -a_i \sin\psi_i \\ \sin\psi_i & a_i \cos\psi_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ \omega_i \end{bmatrix}, \qquad (1)$$

where u_i and ω_i are, respectively, its linear and angular velocities, $\mathbf{h}_i = [x_i \; y_i]^T \in \Re^2$ is the vector containing the coordinates of the point of interest (the point whose position should be controlled), ψ_i is its orientation with respect to the global axis x, and $a_i > 0$ is the distance from the point of interest to the point in the center of the virtual axle of the platform.

Taking into account only the coordinates of the point of interest $\mathbf{h}_i$, the inverse kinematics is

$$\mathbf{v}_i = [u_i \; \omega_i]^T = \mathbf{K}_i^{-1}\dot{\mathbf{h}}_i, \qquad (2)$$

where

$$\mathbf{K}_i^{-1} = \begin{bmatrix} \cos\psi_i & \sin\psi_i \\ -\frac{1}{a_i}\sin\psi_i & \frac{1}{a_i}\cos\psi_i \end{bmatrix}.$$

Therefore, considering a formation of three mobile robots, the forward kinematics of the structure is

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_3 \end{bmatrix},$$

where the subscript $i = 1, \ldots, 3$ stands for the i-th robot. Notice that robots with different kinematic models could be used (heterogeneous formation), changing $\mathbf{K}_i \in \Re^{2\times 2}$ in the matrix $\mathbf{K} \in \Re^{6\times 6}$ accordingly, ever since the nonsingularity of the matrix $\mathbf{K}$ is preserved.

3 The Multilayer Control Scheme

This section briefly describes the multilayer control scheme here adopted, originally proposed in Brandão et al. (2009) to guide a formation of three robots, now extended to guide a multi-robot formation of $n > 3$ robots. Figures 2 and 3 illustrate the flow diagram of the whole structure. The more important layers are the Formation Control layer, the Robot Formation layer, and the Environment layer. Above such layers, a planning layer, the Formation Off-line Planning layer of Fig. 2, can be included. It would be responsible for setting up the initial conditions and the initial positions of the robots in the formation, and for generating either the trajectory to be tracked, or the path to be followed or the target position to be reached.

The Formation Control layer is responsible for generating the control signals sent to the robots in order to reach their desired positions. The Robot Formation layer represents the mobile robots (with differential drive, car-like, and/or omnidirectional kinematic models, as discussed in Sect. 2). Finally, the Environment layer represents all the objects surrounding the robots, including the robots themselves, with their external sensing systems, necessary to implement obstacle avoidance. Fig. 2 illustrates each of the aforementioned layers, while Fig. 3 gives details of the Robot Formation layer. In this layer, three robots composing a triangular structure represent the group #j of the formation. One can notice that each robot has its own dynamic compensation module (Brandão et al. 2009) and its own obstacle avoidance subsystem. Therefore, the present proposal can be a decentralized control and obstacle avoidance strategy for a multi-robot formation, if each robot runs its controller, or a centralized one, if a single computer runs all the controllers.

In the simulations and experiments reported in Sect. 7, a centralized control structure is considered: A single computer runs all the individual controllers (considering the dynamic parameters and controller gains of each robot) and obstacle avoidance modules (considering the information provided by the range finders onboard the robots). For the simulations, as all the models are run in a single computer, there is no com-

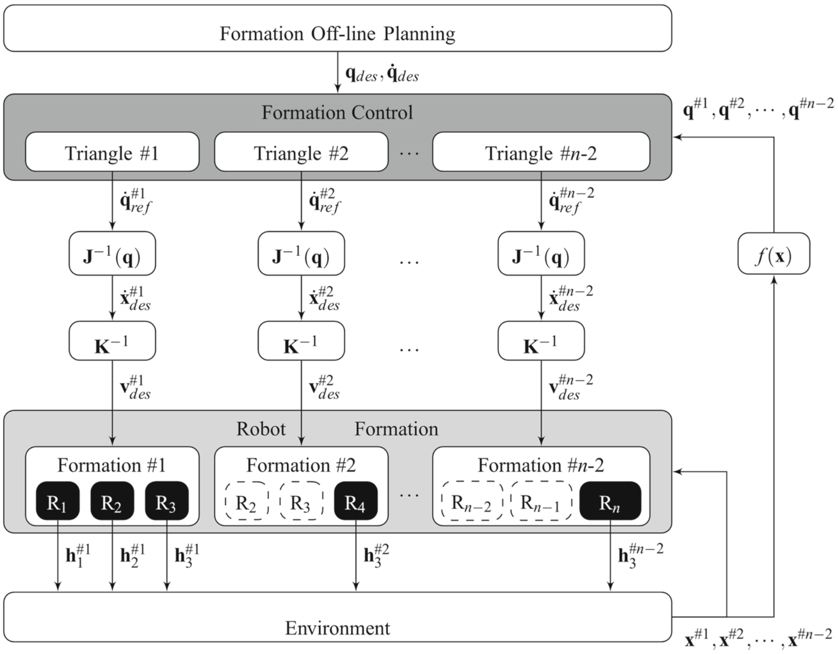

Formation Off-line Planning
$\mathbf{q}_{des}, \dot{\mathbf{q}}_{des}$
Formation Control
Triangle #1
Triangle #2
…
Triangle #n-2
$\mathbf{q}^{\#1}, \mathbf{q}^{\#2}, \cdots, \mathbf{q}^{\#n-2}$
$\dot{\mathbf{q}}^{\#1}_{ref}$
$\dot{\mathbf{q}}^{\#2}_{ref}$
$\dot{\mathbf{q}}^{\#n-2}_{ref}$
$\mathbf{J}^{-1}(\mathbf{q})$
$\mathbf{J}^{-1}(\mathbf{q})$
…
$\mathbf{J}^{-1}(\mathbf{q})$
$f(\mathbf{x})$
$\dot{\mathbf{x}}^{\#1}_{des}$
$\dot{\mathbf{x}}^{\#2}_{des}$
$\dot{\mathbf{x}}^{\#n-2}_{des}$
$\mathbf{K}^{-1}$
$\mathbf{K}^{-1}$
…
$\mathbf{K}^{-1}$
$\mathbf{v}^{\#1}_{des}$
$\mathbf{v}^{\#2}_{des}$
$\mathbf{v}^{\#n-2}_{des}$
Robot Formation
Formation #1
R_1 R_2 R_3
Formation #2
R_2 R_3 R_4
…
Formation #n-2
R_{n-2} R_{n-1} R_n
$\mathbf{h}^{\#1}_1$
$\mathbf{h}^{\#1}_2$
$\mathbf{h}^{\#1}_3$
$\mathbf{h}^{\#2}_3$
$\mathbf{h}^{\#n-2}_3$
Environment
$\mathbf{x}^{\#1}, \mathbf{x}^{\#2}, \cdots, \mathbf{x}^{\#n-2}$

Fig. 2 Flow diagram of the Multilayer control scheme

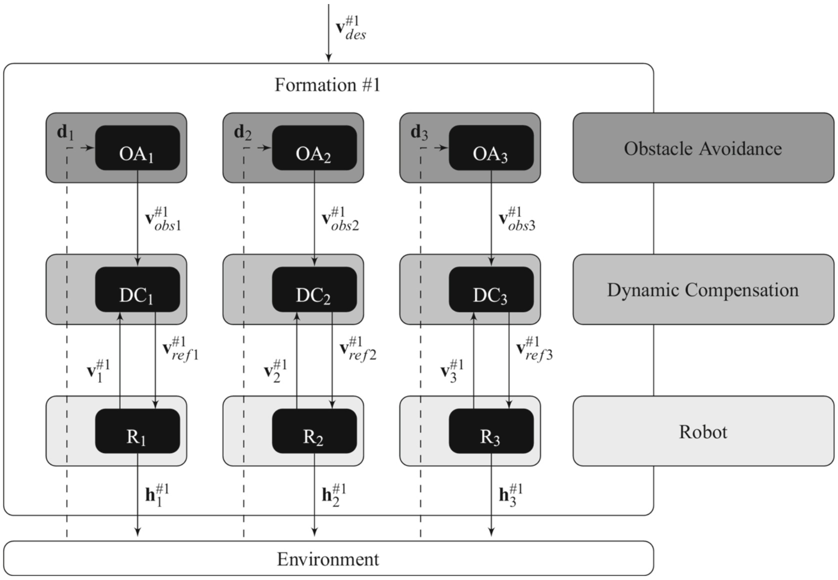


Fig. 3 Detailed view of the Formation #1 in the Robot Formation layer

munication channel established. As for the experiments, the communication channel is an ad hoc local network involving the computers onboard the robots (used to run low-level control and to provide feedback on the robot position and on the distance robot-obstacle) and a fourth computer, running the three controllers and the obstacle avoidance modules.

One advantage of the proposed scheme is that each layer is essentially an independent module, dealing with a specific part of the problem of formation control, in a way that is similar to the one in Fierro and Das (2002). Actually, some layers or even some individual modules inside a layer can be suppressed (e.g., the dynamic compensation module inside the Robot Formation layer could be suppressed, for an application demanding low velocities, or the obstacle avoidance module could be suppressed, if the environment is a strongly structured obstacle-free one). Moreover, the proposed structure is also modular in the horizontal sense, i.e., it grows horizontally for each new robot added to the formation (see Sect. 4.3).

Some additional blocks complete the multilayer scheme, as shown in Fig. 2. They are $\mathbf{J}^{-1}(\cdot)$, the inverse Jacobian matrix, $\mathbf{K}^{-1}$, the inverse kinematic model of the robots, and $f(\cdot)$, the forward kinematic transformation from the robot variables to the formation variables (see Sect. 4).

In this paper, the Off-line Planning layer of Fig. 2 generates a reference trajectory that is directly sent to the Control Layer, besides setting up the initial conditions. In previous works, we have also dealt with the problem of rearranging the assigned initial poses for the individual robots (Rampinelli et al. 2009), which is briefly described in Sect. 5, and with the problem of dynamic compensation (Brandão et al. 2009), aiming at compensating for the dynamics of each robot, thus reducing the velocity tracking error. To deal with the dynamic compensation, a velocity-based dynamic model for differential drive robots was adopted (Martins et al. 2008). As shown in Fig. 3, the k-th group of three robots receives from the Off-line Planning layer, after proper conversion, the d esired velocities $\mathbf{v}^{\#k}_{\text{des}} = [\mathbf{v}^{\#k}_{\text{des1}}\ \mathbf{v}^{\#k}_{\text{des2}}\ \mathbf{v}^{\#k}_{\text{des3}}]^T$, and generates velocity references $\mathbf{v}^{\#k}_{\text{ref}} = [\mathbf{v}^{\#k}_{\text{ref1}}\ \mathbf{v}^{\#k}_{\text{ref2}}\ \mathbf{v}^{\#k}_{\text{ref3}}]^T$ to be sent to the individual robots in the group. Here, $\mathbf{v}^{\#k}_{\text{desi}} = [u^{\#k}_{\text{desi}}\ \omega^{\#k}_{\text{desi}}]^T$ are the desired linear and angular velocities, and $\mathbf{v}^{\#k}_{\text{refi}} = [u^{\#k}_{\text{refi}}\ \omega^{\#k}_{\text{refi}}]^T$ are the reference velocities, both for the i-th robot of the k-th triangle.

If an obstacle is detected close to any robot, its desired linear and angular velocities are changed by the obstacle avoidance module, so that $\mathbf{v}^{\#k}_{\text{des}}$ becomes $\mathbf{v}^{\#k}_{\text{obs}}$ (see Fig. 3). Thus, a free navigation path is guaranteed, as demonstrated in Sect. 7.

4 The Formation Control Layer

This section implements the Control Layer for a centralized formation control considering three or more differential drive mobile robots. The first step is to design a control system considering just a group of three robots. In the sequel, such a control system will be generalized to a n-robot considering $n-2$ triangular groups.

The state variables used to represent a three-robot formation (robots R_1, R_2 and R_3) are shown in Fig. 4. As proposed in Mas and Kitts (2009), the formation pose is given by $\mathbf{P}_{\text{F}} = [x_{\text{F}}\ y_{\text{F}}\ \psi_{\text{F}}] \in \Re^3$, where $(x_{\text{F}}, y_{\text{F}})$ represents the position of the centroid of the formation and ψ_{F} is its orientation with respect to the axis y of the global reference frame xy adopted. In addition, the shape of the formation is given by $\mathbf{S}_{\text{F}} = [p_{\text{F}}\ q_{\text{F}}\ \beta_{\text{F}}] \in \Re^3$, respectively, the distance between R_1 and R_2, the distance between R_1 and R_3, and the angle $R_2\widehat{R_1}R_3$.

4.1 Forward and Inverse Kinematics Transformations

Before introducing the formation control law, it is necessary to express the relationship between the formation pose and shape and the robot positions $\mathbf{h}_i$, which is given by the forward and inverse kinematic relationships, i.e., $\mathbf{q} = f(\mathbf{x})$ and $\mathbf{x} = f^{-1}(\mathbf{q})$, where $\mathbf{q} = [\mathbf{P}_{\text{F}}\ \mathbf{S}_{\text{F}}]^T \in \Re^6$ and $\mathbf{x} = [\mathbf{h_1}\ \mathbf{h_2}\ \mathbf{h_3}]^T \in \Re^6$ (the individual robot orientations are not being considered in this proposal).

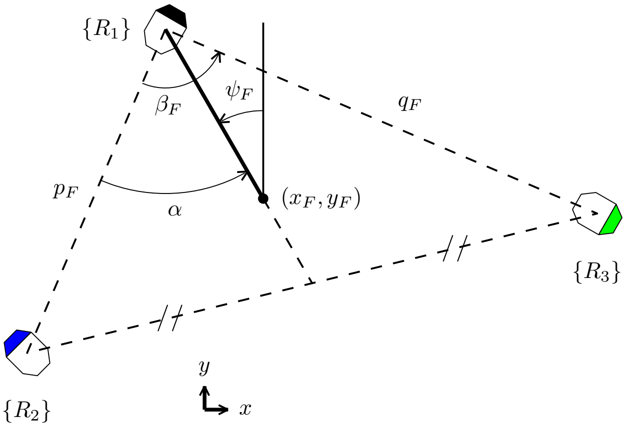


Fig. 4 Formation variables of a sequence **ABC**

The forward kinematic transformation $f(\cdot)$ (see Fig. 4) is given by

$$\mathbf{P}_{\text{F}} = \begin{bmatrix} \dfrac{x_1 + x_2 + x_3}{3} \\ \dfrac{y_1 + y_2 + y_3}{3} \\ \arctan \dfrac{\frac{2}{3}x_1 - \frac{1}{3}(x_2 + x_3)}{\frac{2}{3}y_1 - \frac{1}{3}(y_2 + y_3)} \end{bmatrix}^T, \tag{3}$$

$$\mathbf{S}_{\text{F}} = \begin{bmatrix} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} \\ \arccos \dfrac{p_{\text{F}}^2 + q_{\text{F}}^2 - r_{\text{F}}^2}{2p_{\text{F}}q_{\text{F}}} \end{bmatrix}^T, \tag{4}$$

where $r_{\text{F}} = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}$.

For the inverse kinematic transformation $f^{-1}(\cdot)$, two representations can be dealt with, depending on the sequence of the robots in the triangular formation (clockwise or counterclockwise). Such dispositions are referred to as the $R_1R_2R_3$ or the $R_1R_3R_2$ sequences, hereinafter named **ABC** or **ACB**, respectively. Details can be found in Rampinelli et al. (2009). In this sense, $\mathbf{x} = f^{-1}_{\mathbf{ABC}}(\mathbf{q})$ is given by

$$\mathbf{x} = \begin{bmatrix} x_{\text{F}} + \frac{2}{3}h_{\text{F}} \sin\psi_{\text{F}} \\ y_{\text{F}} + \frac{2}{3}h_{\text{F}} \cos\psi_{\text{F}} \\ x_{\text{F}} + \frac{2}{3}h_{\text{F}} \sin\psi_{\text{F}} - p_{\text{F}} \sin(\alpha + \psi_{\text{F}}) \\ y_{\text{F}} + \frac{2}{3}h_{\text{F}} \cos\psi_{\text{F}} - p_{\text{F}} \cos(\alpha + \psi_{\text{F}}) \\ x_{\text{F}} + \frac{2}{3}h_{\text{F}} \sin\psi_{\text{F}} + q_{\text{F}} \sin(\beta_{\text{F}} - \alpha - \psi_{\text{F}}) \\ y_{\text{F}} + \frac{2}{3}h_{\text{F}} \cos\psi_{\text{F}} - q_{\text{F}} \cos(\beta_{\text{F}} - \alpha - \psi_{\text{F}}) \end{bmatrix}, \tag{5}$$

where $h_{\text{F}} = \sqrt{\frac{1}{2}\left(p_{\text{F}}^2 + q_{\text{F}}^2 - \frac{1}{2}r_{\text{F}}^2\right)}$ is the length of the segment linking $\{R_1\}$ and the segment $\overline{\{R_2\}\{R_3\}}$, passing through $(x_{\text{F}}, y_{\text{F}})$, and $\alpha = \arccos \dfrac{p_{\text{F}}^2 + h_{\text{F}}^2 - \frac{1}{4}r_{\text{F}}^2}{2p_{\text{F}}h_{\text{F}}}$, where

$r_{\mathrm{F}}^2 = p_{\mathrm{F}}^2 + q_{\mathrm{F}}^2 - 2p_{\mathrm{F}}q_F \cos\beta_{\mathrm{F}}$. On the other hand, $\mathbf{x} = f^{-1}_{\mathbf{ACB}}(\mathbf{q})$ is given by

$$\mathbf{x} = \begin{bmatrix} x_{\mathrm{F}} + \frac{2}{3}h_{\mathrm{F}}\sin\psi_{\mathrm{F}} \\ y_{\mathrm{F}} + \frac{2}{3}h_{\mathrm{F}}\cos\psi_{\mathrm{F}} \\ x_{\mathrm{F}} + \frac{2}{3}h_{\mathrm{F}}\sin\psi_{\mathrm{F}} + p_{\mathrm{F}}\sin(\alpha-\psi_{\mathrm{F}}) \\ y_{\mathrm{F}} + \frac{2}{3}h_{\mathrm{F}}\cos\psi_{\mathrm{F}} - p_{\mathrm{F}}\cos(\alpha-\psi_{\mathrm{F}}) \\ x_{\mathrm{F}} + \frac{2}{3}h_{\mathrm{F}}\sin\psi_{\mathrm{F}} - q_{\mathrm{F}}\sin(\beta_{\mathrm{F}}-\alpha+\psi_{\mathrm{F}}) \\ y_{\mathrm{F}} + \frac{2}{3}h_{\mathrm{F}}\cos\psi_{\mathrm{F}} - q_{\mathrm{F}}\cos(\beta_{\mathrm{F}}-\alpha+\psi_{\mathrm{F}}) \end{bmatrix}. \qquad (6)$$

Taking the first time derivative of the forward and the inverse kinematics transformations, the relationship between $\dot{\mathbf{x}}$ and $\dot{\mathbf{q}}$ is represented by the Jacobian matrix $\mathbf{J}(\mathbf{x})$ in the forward way and by the inverse Jacobian matrix $\mathbf{J}^{-1}(\mathbf{q})$ in the inverse way, where

$$\mathbf{J}(\mathbf{x}) = \frac{\partial\mathbf{q}_{n\times1}}{\partial\mathbf{x}_{m\times1}} \quad \text{and} \quad \mathbf{J}^{-1}(\mathbf{q}) = \frac{\partial\mathbf{x}_{m\times1}}{\partial\mathbf{q}_{n\times1}},$$

for $m, n = 6$. Notice that there will be an inverse Jacobian matrix for the **ABC** sequence, and another for the **ACB** sequence, which should be taken into account.

An important issue to be analyzed here is the possible singularities of the Jacobian matrices $\mathbf{J}$ and $\mathbf{J}^{-1}$. They can be singular only in two situations: either when two of the three (or even all the three) robots in the formation are in the same position, or when all of them are aligned (in both cases, there is no triangle anymore). The first situation is not relevant, since it cannot occur in real applications. As for the second situation, it can be prevented by adding a supervisor to the proposed controller, to check whether the angle β_{F} is close to 180 degrees. Case yes, the supervisor should command one of the three robots of the formation to move to a different position in order to decrease β_{F}. Considering that this is done, $\mathbf{J}$ and $\mathbf{J}^{-1}$ are effectively nonsingular (or full rank) matrices.

4.2 The Proposed Control Law

The Formation Control layer receives from the upper layer the desired pose and shape of the formation, $\mathbf{q}_{\mathrm{des}} = [\mathbf{P}_{\mathrm{Fdes}}\ \mathbf{S}_{\mathrm{Fdes}}]^T \in \Re^6$, its desired time derivatives, $\dot{\mathbf{q}}_{\mathrm{des}} = [\dot{\mathbf{P}}_{\mathrm{Fdes}}\ \dot{\mathbf{S}}_{\mathrm{Fdes}}]^T \in \Re^6$, and the current position of all the robots in the formation. Defining the formation error as $\tilde{\mathbf{q}} = \mathbf{q}_{\mathrm{des}} - \mathbf{q}$, the proposed formation control law is

$$\dot{\mathbf{q}}_{\mathrm{ref}} = \dot{\mathbf{q}}_{\mathrm{des}} + \mathbf{L}\tanh\left(\mathbf{L}^{-1}\boldsymbol{\kappa}\tilde{\mathbf{q}}\right), \qquad (7)$$

where $\kappa \in \Re^6$ and $\mathbf{L} \in \Re^6$ are positive definite diagonal gain and saturation matrices, respectively.

Notice that the control signal is saturated through the *tanh* function, to prevent the saturation of the actuators of each robot, guaranteeing that $\dot{\mathbf{q}}_{\mathrm{ref}}$ is below the bounds of the robot actuators. In other words, if there is a large formation error, the saturation function prevents large reference values for the velocities of the individual robots. Thus, the saturation is considered when analyzing the system stability, what would not happen if the control signals were not saturated.

Let $\boldsymbol{\delta}_v$ be the difference between the reference and the real formation variations, or $\boldsymbol{\delta}_v = \dot{\mathbf{q}}_{\mathrm{ref}} - \dot{\mathbf{q}}$. Then, the closed-loop system equation becomes

$$\dot{\tilde{\mathbf{q}}} + \mathbf{L}\tanh\left(\mathbf{L}^{-1}\boldsymbol{\kappa}\tilde{\mathbf{q}}\right) = \boldsymbol{\delta}_v. \qquad (8)$$

Considering the Lyapunov candidate function $V = \frac{1}{2}\tilde{\mathbf{q}}^T\tilde{\mathbf{q}} > 0$, its first time derivative is $\dot{V} = \tilde{\mathbf{q}}^T\dot{\tilde{\mathbf{q}}} = \tilde{\mathbf{q}}^T\left[\boldsymbol{\delta}_v - \mathbf{L}\tanh\left(\mathbf{L}^{-1}\boldsymbol{\kappa}\tilde{\mathbf{q}}\right)\right]$. Observing $\dot{V}$, one cannot conclude immediately about the behavior of the formation error variables. However, assuming that the robots move with low velocities and their masses are not big enough to prevent immediate response to the velocity commands, the vehicle dynamics can be neglected. Thus, one can consider that $\boldsymbol{\delta}_v = \mathbf{0}$, getting $\dot{V} = \tilde{\mathbf{q}}^T\dot{\tilde{\mathbf{q}}} = -\tilde{\mathbf{q}}^T\mathbf{L}\tanh\left(\mathbf{L}^{-1}\boldsymbol{\kappa}\tilde{\mathbf{q}}\right) < 0$. The result is that the equilibrium of the closed-loop system equation is globally asymptotically stable, i.e., $\tilde{\mathbf{q}} \to \mathbf{0}$ when $t \to \infty$.

On the other hand, considering that $\boldsymbol{\delta}_v$ is nonzero, its value is a bounded one if $\dot{\mathbf{q}}_{\mathrm{ref}}$ and $\dot{\mathbf{q}}$ are bounded. Indeed, (7) shows that $\dot{\mathbf{q}}_{\mathrm{ref}}$ is bounded if $\dot{\mathbf{q}}_{\mathrm{des}}$, a user-defined value, is bounded (which is coherent). In addition, a smooth navigation allows, admitting that $\dot{\mathbf{q}}$ is also bounded. Therefore, one can suppose that $\boldsymbol{\delta}_v$ is bounded. In such situation, the equilibrium of (8) will be attractive, taking into account $\dot{V}$, if the negative term is greater then the positive one. In other words, $|\tilde{\mathbf{q}}^T\boldsymbol{\delta}_v| < \tilde{\mathbf{q}}^T\mathbf{L}\tanh\left(\mathbf{L}^{-1}\boldsymbol{\kappa}\tilde{\mathbf{q}}\right)$.

For small control formation errors, one has $\mathbf{L}\tanh\left(\mathbf{L}^{-1}\boldsymbol{\kappa}\tilde{\mathbf{q}}\right) \approx \boldsymbol{\kappa}\tilde{\mathbf{q}}$, and a sufficient condition for $\tilde{\mathbf{q}}$ to decrease is $\|\tilde{\mathbf{q}}\| > \|\boldsymbol{\delta}_v\|/\lambda_{\min}(\boldsymbol{\kappa})$, where $\lambda_{\min}(\boldsymbol{\kappa})$ represents the minimum eigenvalue of $\boldsymbol{\kappa}$. For large control errors one has $|\mathbf{L}\tanh\left(\mathbf{L}^{-1}\boldsymbol{\kappa}\tilde{\mathbf{q}}\right)| \approx \mathbf{L}\mathrm{sign}(\tilde{\mathbf{q}})$, and a sufficient condition for $\tilde{\mathbf{q}}$ to decrease is $\lambda_{\min}(\mathbf{L}) > \|\boldsymbol{\delta}_v\|$, where $\lambda_{\min}(\mathbf{L})$ represents the minimum eigenvalue of $\mathbf{L}$. This means that the formation error $\tilde{\mathbf{q}}$ is ultimately bounded by $\|\boldsymbol{\delta}_v\|/\lambda_{\min}(\boldsymbol{\kappa})$.

In other words, $\tilde{\mathbf{q}}$ decreases until reaching a bound, whose maximum value is given by $\|\boldsymbol{\delta}_v\|/\lambda_{\min}(\boldsymbol{\kappa})$. However, inside the region defined by such bound, it is not possible to define the behavior of the formation error. Nevertheless, one can guarantee that such an error is bounded. Moreover, the bound on such an error can be reduced as $\dot{\mathbf{q}}$ gets closer to $\dot{\mathbf{q}}_{\mathrm{ref}}$ (in the cases of displacements with low velocities, for instance), which falls in the case correspondent to $\boldsymbol{\delta}_v = 0$. In Brandão et al. (2009), it is shown that $\boldsymbol{\delta}_v$ can be reduced by introducing a Dynamic Compensation layer, as illustrated in Fig. 3. This layer implements an adaptive controller, using a robust updating law, to adjust estimated parameters used in the design of the controller in order to better compensate for the dynamics of each robot (Martins et al. 2008). In this situation, the velocity tracking errors will be lowered, for the individual robots, causing a reduction in the formation velocity tracking error $\boldsymbol{\delta}_v$.

4.3 Generalizing the Control Structure for n-Robot Formations

This subsection proposes a way to generalize the control system associated with a formation of three robots (a triangular formation) to a formation of $n > 3$ robots. Such proposition is based on the decomposition of a n-vertices polygon into simpler components, in this case $n - 2$ triangles. The idea is to make profit of the control scheme proposed for a triangular formation to implement a coordinated control of $n > 3$ robots using the same control law presented in Sect. 4.2, thus not demanding to change the Jacobian matrix characterizing the triangular formation. In contrast to other approaches where the complexity of the formation control grows with the number of robots, the complexity of the system proposed here does not increase. Notice that in this case the complexity is associated with the Jacobian matrix, which relates the position of the individual robots to the shape and pose of the formation.

To do that, one should first label the robots (R_i, for $i = 1, \cdots, n$) and determine the leader triangle of the whole formation [$R_2\widehat{R_1}R_3$ or $R_3\widehat{R_1}R_2$, paying attention to the sequence **ABC** or **ACB** (Rampinelli et al. 2009)]. After that, new triangles are formed with the remaining robots, based on a quite simple algorithm: A new triangle is formed with the last two robots of the last triangle already formed and the next robot in the list of labeled robots (i.e., $R_{j+1}\widehat{R_j}R_{j+2}$ or $R_{j+2}\widehat{R_j}R_{j+1}$, where $j = 1, \cdots, n-2$ represents the current triangular formation). Then, each one of such triangles is controlled using the controller designed in Sect. 4.2, as depicted in Fig. 2.

It should be emphasized that no strategy has been adopted to avoid triangle superposition, such as Delaunay triangulation. Notice that triangle superposition is not a problem to our control strategy, because when a new triangle is formed, the previous ones have already been defined, in terms of the formation variables. It even does not matter if the superposition is partial or full. Anyway, to avoid triangle superposition, one needs only to change the robot labeling in the Off-line Planing layer (detailed in Sect. 5).

In addition, the off-line formation planning sets the desired formation variables, $\mathbf{S}_{\mathrm{F}j} = [p_{\mathrm{F}j}\ q_{\mathrm{F}j}\ \beta_{\mathrm{F}j}]$ assigned to each triangular formation. Due to the strategy used to define the triangles, one of the formation variables has its value defined by the previous formation, i.e., $2(n-2)+1$ variables are considered, instead of $3(n-2)$, because it is assumed that $\mathbf{S}_{\mathrm{F}j} = [r_{\mathrm{F}j-1}\ q_{\mathrm{F}j}\ \beta_{\mathrm{F}j}]$ for $j > 1$ (see Fig. 2; Eq. (4)). Furthermore, as depicted in Fig. 2, the $(j+1)$-th controller sends control signals just for one robot (except for the first controller), which is the one not present in the previous triangular formation (the dashed lines in such figure means that the corresponding control signals are inactive.

In terms of system stability, we have proven the convergence of the formation errors relative to each triangle to zero or a bounded value. Considering a certain triangle, for instance, the $(j+1)$-th triangle, two of its vertices go to their desired positions, controlled by the j-th controller. Thus, as the $(j+1)$-th controller guarantees the asymptotic convergence of the third vertex of such triangle to its desired position, dynamically defined from the desired formation initially established, one can perceive that the whole structure will converge to the desired formation, thus allowing conjecturing the stability of the whole control system. Indeed, such a conjecture is supported by simulated and experimental results, some of which are presented in the sequel. However, it should be emphasized that the time response of the controller of each individual robot is faster than the time response of the whole structure. Therefore, as each individual robot reaches its desired position the correspondent triangle becomes correctly composed, and only after all of them reach their positions the entire formation is correctly composed.

Finally, a consequence of the way in which the control scheme is implemented, it is important to stress that if any of the robots in the formation breaks down, the formation should be organized again, defining new triangles, before continuing the navigation. In other words, if one of the agents in the platoon is lost, the other members of the platoon can continue navigating, after the necessary reorganization of the triangular cells.

5 Comments on the Off-Line Planning Layer

The formation off-line planning layer is responsible for generating the desired position to be reached by the formation or the desired trajectory to be followed by the robots during the task accomplishment. In the case where the robots are initially randomly distributed in the environment, such layer is also responsible for optimizing the path to be followed by each individual robot by assigning its desired posture accordingly, to build the formation before starting the task accomplishment. In a previous work (Rampinelli et al. 2009), we have already proposed a solution for the problem of formation rearrangement, which could be also applied if a robot of the formation breaks down.

A suitable pose assignment is quite important, because of the transitory robot positions while taking their initial positions in the formation. Notice that during such transitory intra-formation, collisions could happen, specially for large platoons. Section 6 describes the strategy adopted to avoid collisions when the formation is navigating. However, to prevent the occurrence of collisions during the setting up of the initial formation is also recommended.

The first step to build the initial formation is to consider the sequence of robots in the formation previously defined by the user. Then, depending on the case, the Off-line Planning layer itself can reorganize this sequence, aiming at improving the performance during the maneuvers to build the initial formation. To deal with this problem, we consider two approaches to reorganize the sequence of robots, starting from the initial assignment defined by the user.

First, all randomly distributed robots receive a label given by the user. Then, the algorithm named *formation rearrangement through final reorganization* identifies the most distant robot of the desired static formation and label it as R_1. The next robot to be labeled will be the one closest to the last labeled one, and so on, until all robots are labeled. This approach is suggested for formations in which all robots have similar sensorial systems and actuators (load transportation and escorting missions, for instance). Thus, the desired position of each robot can be changed aiming at optimizing the transient arrangement of the platoon and the energy spent during the initial reconfiguration, without affecting the accomplishment of the task.

The second algorithm, which can be seen as a second step of the rearranging algorithm previously described, consists in associating the R_1 robot of the optimized initial formation to the robot in the initial formation labeled by the user that is closest to it. After identifying R_1 in the initial formation, the other robots are labeled using a criterion of proximity (using the Euclidian distance), and respecting the established sequence for the triangle formation (**ABC** or **ACB**) during the rearrangement.

It is worth mentioning that by using one of these approaches, or even considering the configuration proposed by the user, the resulting desired disposition in the formation should be reached by the robots that are initially randomly distributed in the workspace, thus building the initial formation (before starting the coordinated navigation).

6 The Obstacle Avoidance Module

This section describes how the system processes the sensorial information about the surrounding environment (each robot in the formation has an onboard sensing system to allow measuring the distance to obstacles in a certain range), and the way such information is modeled and integrated in the formation control to give to the formation the capability of avoiding obstacles.

The main idea is to associate the movement of each individual robot to virtual forces characterizing its interaction with the surrounding environment. This way, the linear and angular velocities of each robot in the formation will be affected by virtual repulsion forces given by

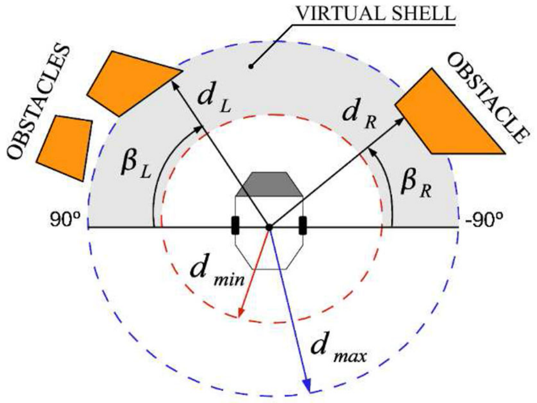


Fig. 5 Interaction between a robot and obstacles in the surrounding environment during navigation

$$F_{\mathrm{F}} = \begin{cases} b(d_{\max} - d)^2, & \text{if} \quad d \in [d_{\min}, d_{\max}] \\ 0, & \text{if} \quad d > d_{\max} \end{cases} \tag{9}$$

which depends on d, the distance from the robot to the closest obstacle, $d_{\min}$, the minimum acceptable robot-obstacle distance to avoid a crash, and $d_{\max}$, the maximum robot-obstacle distance that causes a nonzero fictitious force. The constant b corresponds to system calibration and is given by

$$b(d_{\max} - d_{\min})^2 = F_{f\,\max}. \tag{10}$$

In order to make possible for a multi-robot system to move in environments containing multiple obstacles, each robot actually considers two fictitious forces, which are associated with the minimum robot-obstacle distances at the right and left sides, as illustrated in Fig. 5. The variables d_{R}, d_{L}, β_{R}, and β_{L} represent, respectively, the minimum distances to obstacles in the right and in the left sides of the robot, and their corresponding angles.

Thus, when a robot detects an obstacle within its virtual shell (see Fig. 5), the control signals generated by the formation controller will be adjusted to reduce the linear velocity (except if the obstacle is parallel to the robot) and to modify the angular velocity to turn the robot in the opposite direction, considering the detected obstacle.

To implement this behavior, the linear and angular velocities effectively sent to the robots of the formation are, respectively,

$$u_{\mathrm{obs}} = u_{\mathrm{des}} - u_{\mathrm{cor}} \tag{11}$$

$$\omega_{\mathrm{obs}} = \omega_{\mathrm{des}} + \omega_{\mathrm{cor}}, \tag{12}$$

where u_{des} and ω_{des} are the desired linear and angular velocities, and u_{cor} and ω_{cor} are the corrections added to the linear and angular velocities. Such corrections are the solutions of

$$I_u \ddot{u}_{\mathrm{cor}} + B_u \dot{u}_{\mathrm{cor}} + K_u u_{\mathrm{cor}} = F_{\mathrm{R}} \sin \beta_{\mathrm{R}} + F_{\mathrm{L}} \sin \beta_{\mathrm{L}}, \tag{13}$$

$$I_\omega \ddot{\omega}_{\mathrm{cor}} + B_\omega \dot{\omega}_{\mathrm{cor}} + K_\omega \omega_{\mathrm{cor}} = F_{\mathrm{R}} - F_{\mathrm{L}}, \tag{14}$$

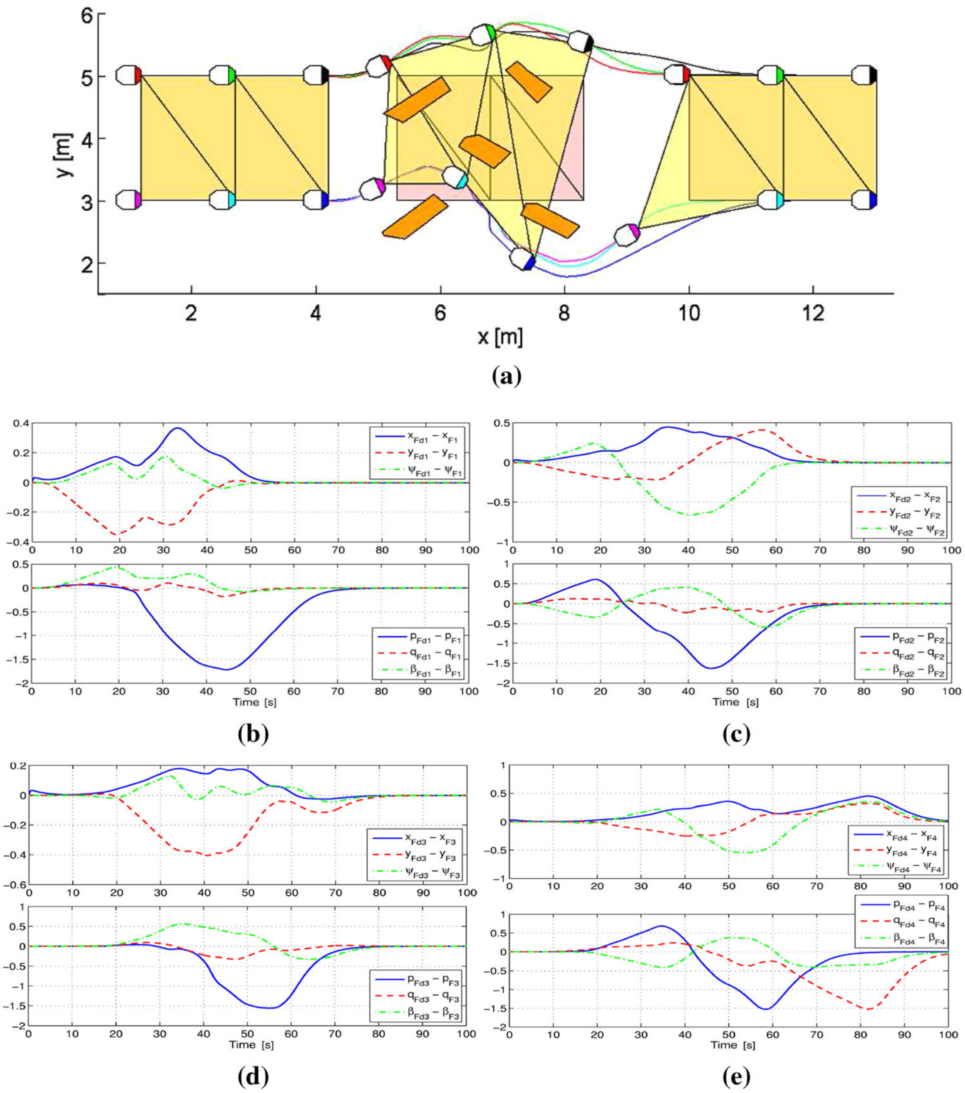


Fig. 6 Simulated trajectory tracking with a platoon of six robots passing through a set of obstacles. **a** Trajectory followed by the formation. **b** Shape and pose errors (Triangle 1). **c** Shape and pose errors (Triangle 2). **d** Shape and pose errors (Triangle 3). **e** Shape and pose errors (Triangle 4)

where F_{R} and F_{L} are the force values due to the right ($\beta_R \in [-90, 0]$ degrees) and left ($\beta_{\text{L}} \in (0, 90]$ degrees) obstacles, respectively. As result, the effect of the presence of obstacles close to the robots in the formation is an adaptation of their velocities, causing a change in the formation shape, necessary to left the obstacles behind. In the Laplace domain, (13) and (14) can be written as

$$u_{\text{cor}} = Z_{\text{u}}^{-1}(F_{\text{R}} \sin \beta_{\text{R}} + F_{\text{L}} \sin \beta_{\text{L}}),$$
$$\omega_{\text{cor}} = Z_{\omega}^{-1}(F_{\text{R}} - F_{\text{L}}),$$

where Z_u and Z_ω represent the mechanical impedance characterizing the robot–environment interaction and are $Z_u = I_u s^2 + B_u s + K_u$ and $Z_\omega = I_\omega s^2 + B_\omega s + K_\omega$, respectively. The terms I_u and I_ω, B_u and B_ω, and K_u and K_ω are positive constants representing, respectively, the effect of the inertia, the damping and the elastic constant, as proposed in Hogan (1985). In the context of this work, Z_u and Z_ω also represent a repulsive dynamic gain of the linear velocity and the lever of the angular velocity.

In opposition to the classical impedance method, the obstacle avoidance strategy here proposed consists in disturbing the velocity commands sent to each robot individually. Notice also that such strategy is applied to each robot individually, as depicted in Figs. 3 and 5, thus not increasing the computational effort when the number of robots in the formation increases. On the other hand, if the formation is supposed to navigate in an obstacle-free environment, such module can be simply discarded.

An important aspect associated with the way the angles β_{L} and β_{R} are defined is that in the case of an obstacle just in front of the robot, only a force F_{R} is generated (due to the definition of β_{R} and β_{L}), thus causing a rotation of the robot, avoiding such obstacle.

A final remark regarding the system stability is that during the obstacle avoidance procedure the tracking error $\boldsymbol{\delta}_v$ will increase due to the modification of the velocity commands

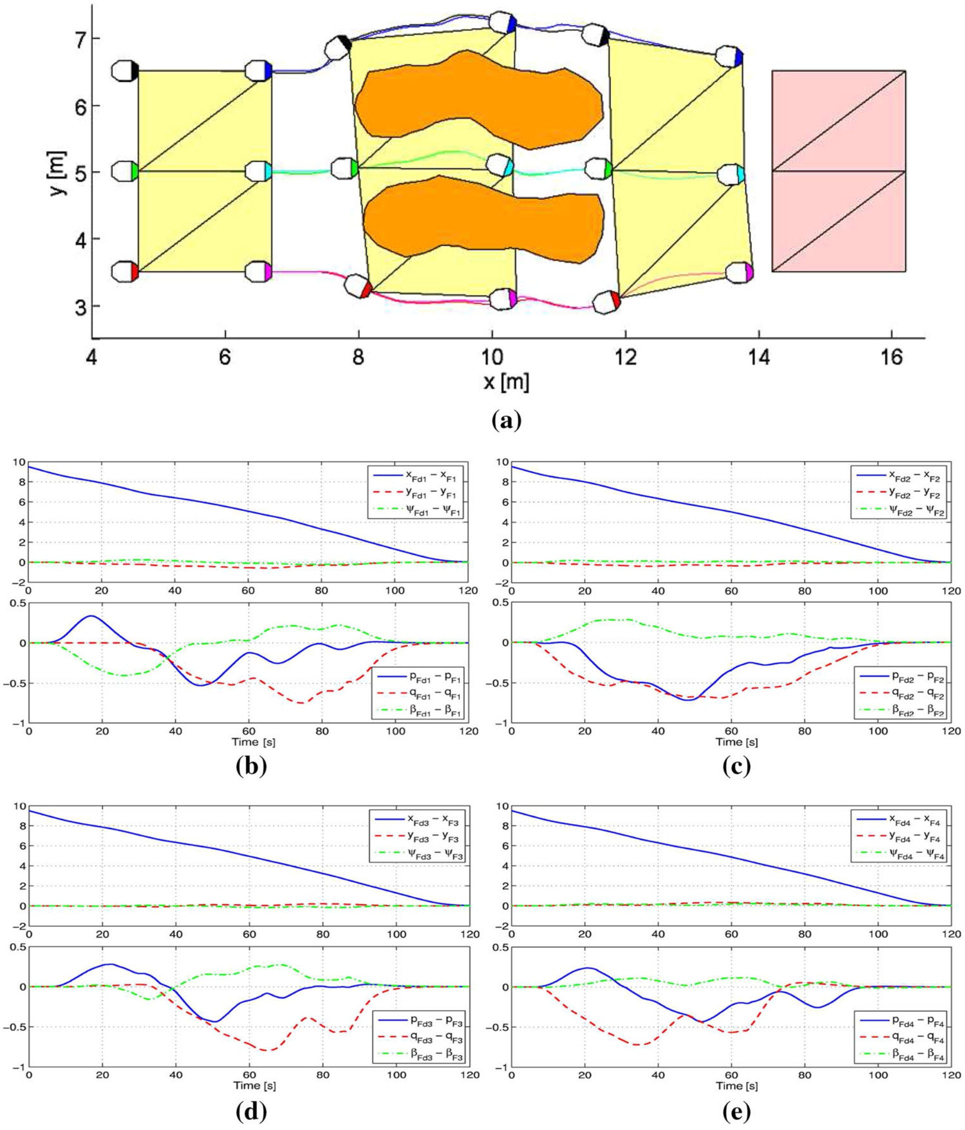


Fig. 7 Simulated positioning control with a platoon of six robots passing through a corridor obstacle. **a** Path followed by the formation. **b** Shape and pose errors (Triangle 1). **c** Shape and pose errors (Triangle 2). **d** Shape and pose errors (Triangle 3). **e** Shape and pose errors (Triangle 4)

caused by the fictitious forces. However, once the fictitious forces become null (i.e., the robots have left the obstacles behind), the velocity commands restart being sent directly by the formation controller (7), so that the stability analysis presented in Sect. 4.2 is still valid. Thus, the obstacle avoidance module causes a transient increase in the control errors while the obstacles are within the robots range, but once they have been left behind the control errors decrease, showing that the system stability is not affected. Actually, such changes in the velocity commands can be thought as disturbances to the control system.

7 Results and Discussion

In this section, four examples are presented, in order to check the effectiveness of the control system proposed to guide a multi-robot formation. Two of them are simulations, performed using a proprietary simulator that uses the model of the robot Pioneer 3-DX (from Adept Mobile Robots) for all the robots in the formation, for which the model of the SICK 2D laser scanner is adopted as the onboard rangefinder. As for the other two examples, they are real experiments run using two Pioneer 3-AT robots, both with a SICK 2D laser scanner onboard, and one Pioneer 3-DX robot, with an array of ultrasonic sensors in the front, all from Adept Mobile Robots.

As for the parameters adopted for the obstacle avoidance module, they were $b = 1$, $d_{\min} = 0.30\,\text{m}$, $d_{\max} = 0.95\,\text{m}$, $I_u = 0.4s^3$, $B_u = 5.6s^2$, $K_u = 10s$, $I_\omega = 0.2s^3$, $B_\omega = 1.4s^2$, and $K_\omega = 0.65s$, for the simulations and the experiments. By their turn, the gain and saturation matrices adopted are $\boldsymbol{\kappa} = 0.25\mathbf{I}$ and $\mathbf{L} = 0.10\mathbf{I}$, for the first simulation and the two experiments (trajectory tracking tasks). As for the second simulation, a case of positioning, the values selected

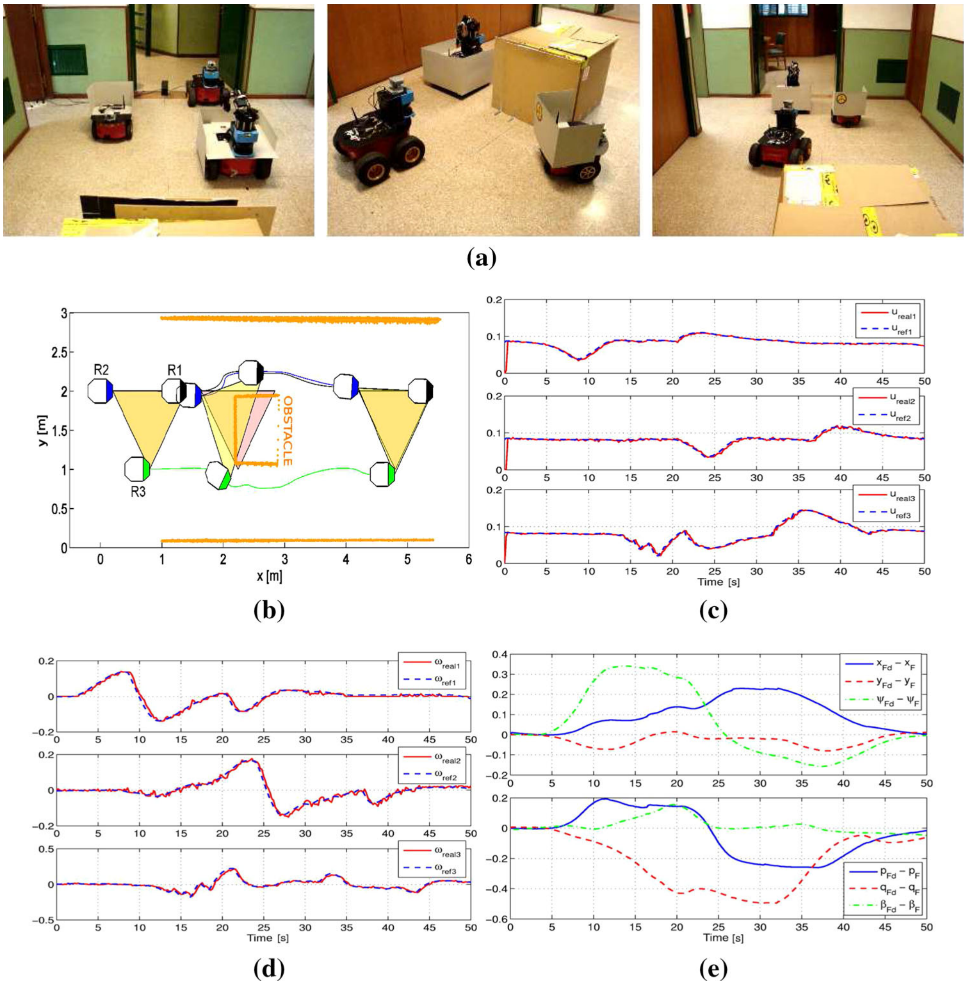


Fig. 8 Results for an experiment with three robots, in which an obstacle is in the middle of a corridor. **a** Three snapshots of the first experiment. **b** Recovered trajectories of the individual robots and their positions in connection with the snapshots shown in a. **c** Commanded and real linear velocities. **d** Commanded and real angular velocities. **e** Temporal evolution of the formation errors

were $\boldsymbol{\kappa} = 0.12\mathbf{I}$ and $\mathbf{L} = 0.20\mathbf{I}$. In all cases, $\mathbf{I} \in \Re^{6\times 6}$ is an identity matrix.

7.1 Simulated Results

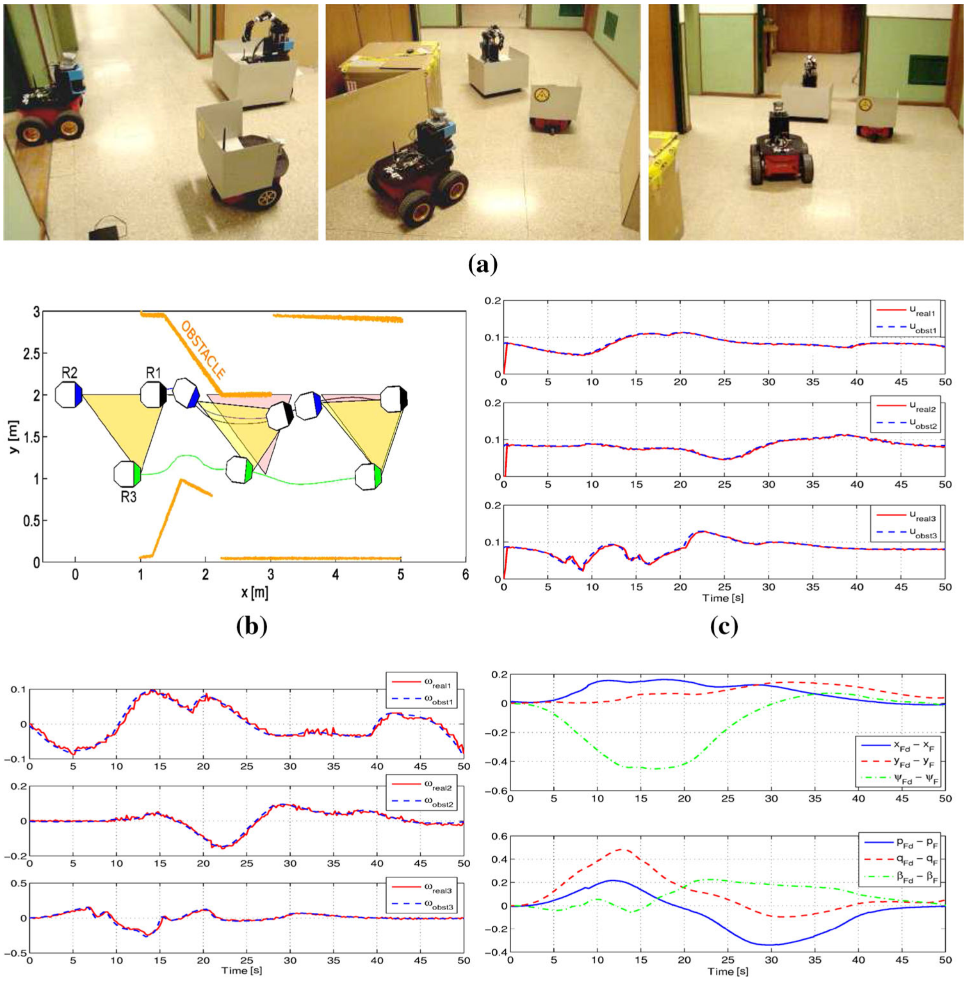
In the first simulation, a platoon of six robots should follow a horizontal line, from left to right, in a partially structured environment containing five obstacles in the planned trajectory. The six robots are supposed to avoid the obstacles preserving at most the original formation, which they should resume after leaving the obstacles behind. By its turn, the second simulation corresponds to a positioning task, in which the same six-robot platoon should avoid obstacles on its path when navigating toward a target formation (this could be the case of mounting the initial formation before starting the navigation to accomplish a given task, for instance).

Figures 6a and 7a show three particular instants of such simulations. In the left side of such figures, one can see the initial formation. After some time of navigation, as depicted in the middle of such figures, the formation is distorted, since the individual robots are avoiding the obstacles (thus, the formation is temporarily lost). Finally, as shown in the right part of such figures, after some time of navigation, the formation shape is resumed, since the robots have left the obstacles behind. As it can be noticed, no collision is produced at any time during the simulations, and the formation shape is recovered after leaving the obstacles behind.

The formation pose and shape errors for the four triangular groups (depicted in Figs. 6a, 7a) are shown in Figs. 6b–e and 7b–e, respectively, for the two simulations just mentioned. As it can be seen, nonzero formation errors arise during the obstacle avoidance interval (20−60 s), and tend to zero after the robots left the obstacles behind.

Notice that Figure 6 corresponds to a trajectory tracking task, whereas Figure 7 corresponds to a positioning one. The objective here is to show that the proposed controller is able to guide the formation in these two kinds of task, to tell something about its versatility.

As for how the computer running the simulations gets the positions $\mathbf{h}_i$ of the individual robots and their range measurements $\mathbf{d}_i$, the positions are gotten using dead reckoning, with the velocities available from the code, starting from the initial positions of the individual robots, whereas the range mea-



(a)

(b)

(c)

(d)

(e)

Fig. 9 Results for an experiment with three robots, in which obstacles are narrowing the passage along a corridor. **a** Three snapshots of the second experiment. **b** Recovered trajectories of the individual robots and their positions in connection with the snapshots shown in a. **c** Commanded and real linear velocities. **d** Commanded and real angular velocities. **e** Temporal evolution of the formation errors

surements are provided by the models of the laser rangefinder adopted, considering the simulated environment.

7.2 Experimental Results

To run the two experiments, the whole system was programmed in a computer onboard one of the three robots used, to guide the triangular formation to track a trajectory (the formation should go ahead, following a straight line). The other two robots run just the low-level control and sensing system, to drive the controlled wheels and to get the pose and the range measurements to be fed back to the formation controller. As stated before, the communication between the three robots is done through an ad hoc local network involving the three onboard computers (a centralized control system is implemented, with the formation controller running in a computer onboard one of the robots in the formation). Through such communication channel, the computer running the formation control receives the feedback from the individual robots and generates the velocities of reference for them. The scenario of the experiment is a narrow corridor approximately 3 m wide and 7 m long, having obstacles in the trajectory to be followed by the formation.

In the first experiment, there is an obstacle in the middle of the corridor along which the formation should navigate. Fig. 8a shows three snapshots correspondent to three different instants during the experiment run, whereas Fig. 8b depicts the robots in their trajectories in such time instants. To complete the analysis of this experiment, Fig. 8c–e show the linear and angular velocities developed by the robots during the experiment and the correspondent formation errors. As one can see in such figures, the position errors are meaningful only during the obstacle deviation, and after the robots left the obstacles behind, they converge to zero. Thus, the formation is preserved while no obstacle is detected close to any robot of the formation, is temporarily lost during obstacle deviation, and is resumed right after the obstacles are left behind by the robots of the formation.

In the second experiment, the formation composed by the same three robots should navigate along the same corridor, now with obstacles in both sides, narrowing the passage available for the formation. Similarly to Figs. 8a–e, 9a–e illustrate this experiment and exhibit its results.

As one can see by analyzing the correspondent figures, the system effectively performed as expected, in the simulations and in the real experiments as well, thus validating the proposed control and obstacle avoidance systems.

As for the way the robots get their positions $\mathbf{h}_i, i = 1, \ldots, 3$, they use dead reckoning. Each individual robot gets its position, related to its own frame, and the robot running the formation controller receives such information and generates the positions of the other robots with respect to its frame, knowing the initial position of each robot in the formation. The individual robots also deliver their range measurements $\mathbf{d}_i, i = 1, \ldots, 3$, to the one running the controller, since it is also responsible for running the obstacle avoidance modules. As stated before, such communication is done through an ad hoc local networking involving the three onboard computers.

To conclude the description of the experiments, it is worth mentioning that only a formation of three robots was used in the experiments because just three robots were available. However, as these experiments validate the control system correspondent to the first triangular cell, which is the kernel for the generalized formation control system here proposed, we claim that they also validate the whole control scheme.

An important feature associated with the obstacle avoidance subsystem is that it acts only over the specific robot to which it is connected (see Fig. 3). This way, there is no occlusion, in the sense that each individual robot only manages its local navigation, since the control strategy is a centralized one (a central computer receives the sensorial data of each robot and generates the respective control signals, including the obstacle avoidance module). In other words, an individual robot does not "see" the other ones. Actually, when they are close, they are dealt with as obstacles.

8 Conclusion

This paper proposes a multilayer scheme to control a group of any number of robots navigating in a coordinated formation. The proposed control system is able to guide the formation when accomplishing positioning or trajectory tracking tasks, without colliding to any object appearing in its path. The basic assumption is that the formation can be deformed (either expanded or contracted) when avoiding obstacles, which is compatible with a certain group of tasks, like surveillance in large areas, for instance.

In its essence, the proposed control system is a noncentralized one, since a controller is associated to each robot, except for the three first robots, which are governed by a single controller. In addition, the velocities of the individual robots in the formation are changed accordingly, whenever they detect obstacles close to them, so that each robot avoids any obstacle in its path (even another robot of the formation, when it gets too close), thus guaranteeing that the entire formation avoids the obstacles in the environment surrounding it. Finally, results of two simulations and two real experiments are presented, which validate the proposed control system.

References

Antonelli, G., Arrichiello, F., & Chiaverini, S. (2008). The entrapment/escorting mission. *IEEE Robotics & Automation Magazine*, *15*(1), 22–29.

Balch, T., & Arkin, R. C. (1998). Behavior-based formation control for multiagent robot teams. *IEEE Transactions on Robotics and Automation*, *14*(6), 926–934.

Brandao A. S., Sarcinelli-Filho, M., Carelli, R., & Bastos-Filho, T. F. (2009). Decentralized control of leader–follower formations of mobile robots with obstacle avoidance. In *Proceedings of the 5th IEEE international conference on mechatronics (ICM'09), Málaga, Spain*.

Brandão, A. S., Martins, F. N., Rampinelli, V. T. L., Sarcinelli-Filho, M., Bastos-Filho, T. F., & Carelli, R. (2009). A multi-layer control scheme for multi-robot formations with adaptive dynamic compensation. In *Proceedings of the 5th IEEE international conference on mechatronics (ICM'09), Málaga, Spain*.

Cao, Y. U., Fukunaga, A. S., & Kahng, A. B. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, *4*, 7–27.

Chen, Y. Q. & Wang, Z. (2005). Formation control: A review and a new consideration. In *Proceedings of the 2005 IEEE/RSJ international conference on intelligent robots and systems (IROS'05), Edmonton, Canada* (pp. 3181–3186).

Consolini, L., Morbidi, F., Prattichizzo, D., & Tosques, M. (2007). A geometric characterization of leader–follower formation control. In *Proceedings of the 2007 IEEE international conference on robotics and automation (ICRA'07), Rome, Italy* (pp. 2397–2402).

Cruz, D., McClintock, J., Perteet, B., Orqueda, O., Cao, Y., & Fierro, R. (2007). Decentralized cooperative control: A multivehicle platform for research in networked embedded systems. *IEEE Control Systems Magazine*, *27*(3), 58–78.

de La Cruz, C., & Carelli, R. (2008). Dynamic model based formation control and obstacle avoidance of multi-robot systems. *Robotica*, *26*(3), 345–356.

De-Gennaro, M., & Jadbabaie, A. (2006). Formation control for a cooperative multi-agent system using decentralized navigation functions. In *Proceedings of the 2006 American Control Conference (ACC'06), Minneapolis, USA*.

Dong, W., Guo, Y., & Farrell, J. (2006). Formation control of nonholonomic mobile robots. In *Proceedings of the 2006 American control conference (ACC'06), Minneapolis, USA*.

Fierro, R., & Das, A. (2002). A modular architecture for formation control. In *Proceedings of the third international workshop on robot motion and control (RoMoCo'02), Bukowy Dworek, Poland* (pp. 285–290).

Gava, C., Vassallo, R., Roberti, F., Carelli, R., & Bastos-Filho, T. (2007). Nonlinear control techniques and omnidirectional vision for team formation on cooperative robotics. In *Proceedings of the 2007 IEEE international conference on robotics and automation (ICRA'07), Rome, Italy* (pp. 2409–2414).

Golkar, M., Namin, S. T., & Aminaiee, H. (2009). Fuzzy controller for cooperative object pushing with variable line contact. In *Proceedings of the 5th IEEE international conference on mechatronics (ICM'09), Malaga, Spain*.

Hess, M., Saska, M., & Schilling, K. (2009). Application of coordinated multi-vehicle formations for snow shoveling on airports. *Intelligent Service Robotics*, *2*, 205–217.

Hogan, N. (1985). Impedance control: An approach to manipulation. *ASME Journal of Dynamic Systems, Measurement, and Control*, *107*, 1–23.

Hougen, D., Benjaafar, S., Bonney, J. C., Budenske, J., Dvoraktt, M., Gini, M., French, H., Krantz, D., Li, P., Malver, F., Nelson, B., Papanikolopoulos, N., Rybski, P., Stoeter, S., Voyles, R., & Yesin, K. (2000). A miniature robotic system for reconnaissance and surveillance. In *Proceedings of the 2000 IEEE international conference on robotics and automation (ICRA'00), San Francisco, USA* (pp. 501–507).

Jennings, J. S., Whelan, G., & Evans, W. F. (1997). Cooperative search and rescue with a team of mobile robots. In *Proceedings of the 8th international conference on advanced robotics (ICAR'97), Monterrey, USA*, (pp. 193–200).

Jia, Q., Li, G., & Lu, J. (2006). Formation control and attitude cooperative control of multiple rigidbody systems. In *Proceedings of the 60th international conference on intelligent systems design and applications (ISDA'06), Jinan, China* (vol. 2, pp. 82–86).

Liu, B., Zhang, R., and Shi, C. (2006). Formation control of multiple behavior-based robots. In *Proceedings of the international conference on computational intelligence and security (CIS'06), Springer, Guangzhou, China* (vol. 1, pp. 544–547).

Martins, F. N., Celeste, W. C., Carelli, R., Sarcinelli-Filho, M., & Bastos-Filho, T. F. (2008). An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice*, *16*, 1354–1363.

Mas, I., & Kitts, C. (2009). Cluster space specification and control of mobile multirobot systems. *IEEE/ASME Transactions on Mechatronics*, *14*(2), 207–218.

Ogren, P., & Leonard, N. (2003). Obstacle avoidance in formation. In *Proceedings of the 2003 IEEE international conference on robotics and automation (ICRA'03), Taipei, China* (vol. 2, pp. 2492–2497).

Rampinelli, V. T. L., Brandão, A. S., Martins, F. N., Sarcinelli-Filho, M., & Carelli, R. (2009). A multi-layer control scheme for multi-robot formations with obstacle avoidance. In *Proceedings of the 14th International Conference on Advanced Robotics (ICAR'09), Munich, Germany*.

Shao, J., Xie, G., Yu, J., & Wang, L. (2005). Leader-following formation control of multiple mobile robots. In *Proceedings of the 2005 IEEE international symposium on intelligent control (ISIC'05), Limassol, Cyprus* (pp. 808–813).

Stoeter, S. A., Rybski, P. E., Stubbs, K. N., McMillen, C. P., Gini, M., Hougen, D. F., et al. (2002). A robot team for surveillance tasks: Design and architecture. *Robotics and Autonomous Systems*, *40*(2–3), 173–183. N.

Stouten, B., & de Graaf, A. (2004). Cooperative transportation of a large object-development of an industrial application. In *Proceedings of the IEEE international conference on robotics and automation (ICRA'04), Ne Orleans, USA* (vol. 3, pp. 2450–2455).