Robotica (2011) volume 29, pp. 335–349. © Cambridge University Press 2010
doi:10.1017/S0263574710000068

Formation control and trajectory tracking of mobile robotic systems – a Linear Algebra approach

Andrés Rosales*, Gustavo Scaglia, Vicente Mut
and Fernando di Sciascio

Instituto de Automática (INAUT). Universidad Nacional de San Juan, Av. Libertador San Martín 1109 (oeste) – J5400ARL, San Juan, Argentina

(Received in Final Form: February 11, 2010. First published online: May 5, 2010)

SUMMARY

A novel approach for trajectory tracking of a mobile-robots formation by using linear algebra theory and numerical methods is presented in this paper. The formation controller design is based on the formation states concept and the dynamic model of a unicycle-like nonholonomic mobile robot. The proposed control law designed is decentralized and scalable. Simulations and experimental results confirm the feasibility and the effectiveness of the proposed controller and the advantages of using the dynamic model of the mobile robot. By using this new strategy, the formation of mobile robots is able to change its configuration (shape and size) and follow different trajectories in a precise way, minimizing the tracking and formation errors.

KEYWORDS: Dynamic model; Formation states; Linear algebra; Multi-robot system; Numerical methods; Trajectory tracking.

1. Introduction

There is a strong development of applications for the coordinate control of a group of robots. Applications of autonomous vehicle formations range from automated highway systems, where groups of autonomous cars are able to travel safely to formations of unmanned aerial vehicles performing search and rescue operations over dangerous regions. Both the Air Force and NASA have identified autonomous formations of aircraft and spacecraft as key technological milestones for the 21st century.[21] Other examples of tasks that require the cooperation of several robots are robot teams for games, surveillance operations, exploration, survey and mapping, and tasks that need organized robot teams in specific formations, such as moving objects, cooperative handling of objects, etc.[24]

The research on coordinated robots starts after the introduction of the behavior-based control paradigm.[5] Basically, there are three approaches for robot coordination reported in the literature: leader tracking,[8,12,23] behavior methods,[2,16,34] and virtual structure techniques.[4,28,36] Most of the proposed coordination control systems are not based on dynamic systems, principally on account of the complexity of multi-robot systems. This is more so when considering nonholonomic mobile robots. This paper proposes working with the dynamic model of the mobile robot, because this characteristic allows the controller to face up to sudden velocity changes and to improve the performance of the system.[7]

The control task could be either centralized when there is monitoring and control of all robots to place them at the desired position, or decentralized when there is no supervisor and the feedback is only the detected relative position of each robot respecting its neighbor robots.[6] The centralized formation control could represent a good strategy for a small team of robots, when it is implemented with a single computer and a single sensor to monitor and control the whole team. However, when considering a team with a large number of robots, the need of greater computational capacity and a large communication bandwidth could make advisable to use the decentralized formation control. Yamaguchi, *et al.*[38] presented a distributed control scheme and showed simulations for final static formations. Fierro, *et al.*[15] have proposed a hierarchical control structure that allows the switching of controllers in order to have a stable formation, based on sensing their relative positions to neighboring robots, under a strategy of distributed control.

Most of the proposed coordination control systems are not based on dynamic systems, largely on account of the complexity of multi-robot systems. For instance, in Lawton *et al.*[21], the dynamic model of the mobile robot is linearized output feedback and the formation control is designed using passivity techniques, even though the control is based on a sequence of formations and not on a continuous tracking, and the control actions are the forces and the torques which are not simple to apply to real actuators. Das *et al.*[9] designed an input–output feedback linearization controller and used a switching strategy to control formations, although there is a constraint that the leader velocity is bounded away from zero. Antonelli and Chiaverini,[1] introduced a kinematic control of a platoon of autonomous vehicles, with the objective to control the values taken by some task functions or platoon-level functions, such as the average position of the vehicles in a platoon or the distribution of the vehicles about the average position. In the present work these values are called formation states.

* Corresponding author. E-mail: androsaco@gmail.com

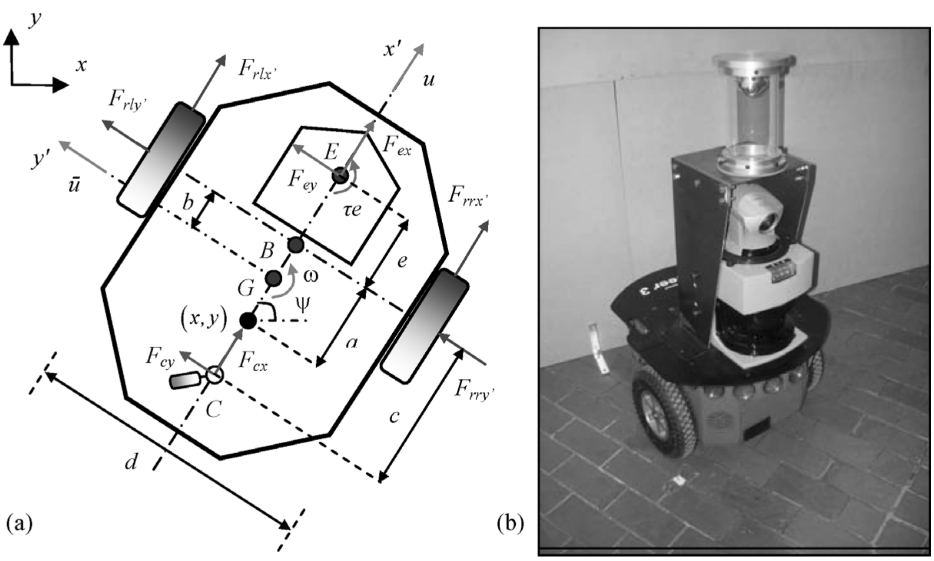

Fig. 1. PIONEER 3DX mobile robot. (a) Model; (b) laboratory equipment.

The use of path tracking in a navigation system is justified in structured workspaces as well as in partially structured workspaces where unexpected obstacles can be found during navigation.[18] In the first case, the reference trajectory can be set from a global trajectory planner. In the second case, algorithms used to avoid obstacles replan the trajectory in order to avoid a collision;[19] then, a new reference trajectory, which must be followed by the robot, is generated. Furthermore, there are algorithms that express the reference trajectory of the mobile robot as a function of a descriptor called r,[11] or s (called "virtual time" [22]), which is a function of the tracking error and the time t. For example, if the tracking error is large, the reference trajectory should wait for the mobile robot; on the other hand, if the tracking error is small, the reference trajectory must tend to the original trajectory calculated by the global planner. Accordingly, the module of trajectory tracking will use the original path or the on-line recalculated path as reference to obtain the smallest error when the mobile robot follows the path.[26] Consequently, path tracking is always important and independent from whether the reference trajectory has been generated by a global trajectory planner or a local one. Most papers about the trajectory tracking problem[13,14,17,20,37,39] have interesting solutions for this theme.

The aim of this paper is to use the linear algebra theory and numerical methods to compute control actions so that the multi-robot formation achieves a position (x, y) with a pre-established orientation ψ at each sample time (kTo). To achieve this objective (e.g., with nonholonomic vehicles), we have two control variables: the linear velocity u and the rotational velocity ω of each robot. The proposed controller, based on the formation states and the dynamic model of a nonholonomic mobile robot, computes the optimal control action (according to least squares),[35] which allows the mobile-robots formation to go from the actual state to the desired one. This approach has been tested in some works.[29,31–33] Simulations and experimental results have been applied to PIONEER 3DX mobile robots. The efficacy and feasibility are then demonstrated in a practical sense through a set of experiments where the speed-range is similar to the one reported in other papers about trajectory tracking.[25]

This paper is organized as follows: Section 2 describes the dynamic model of a mobile robot. Section 3 presents the concept of formation states. The methodology to compute the linear algebra-based controller is described in Section 4, and Section 5 presents simulations and experimental results using the proposed controller on PIONEER 3DX mobile robots. Finally, conclusions are detailed at the end.

2. Dynamic Model of a Nonholonomic Mobile Robot

To perform tasks with requirements of high speed and/or transport of heavy loads, it is very important to consider the dynamics of the mobile robot, because such tasks exert very large external forces on the robot and will inevitably influence its path and direction. Thus, a kinematic model is not sufficient. Dynamic characteristics of the robot, such as mass and inertia center, change if the robot is loaded. Previous studies very often ignored the dynamics of mobile robots and also suffered from algorithmic singularity. A nonholonomic dynamic model of a unicycle-like mobile robot is shown in Fig. 1(a) and is presented in (1).

The robot position is defined by (x, y); this point is located at a distance a from rear axis center of the robot; u and $\bar{u}$ are the longitudinal and side speeds of mass center, respectively; ω is the angular speed; ψ is the orientation angle; G is the gravity center; and B is the base line center of the wheels.

$F_{rrx'}$ and $F_{rry'}$ are the longitudinal and lateral tire forces of the right wheel; $F_{rlx'}$ and $F_{rly'}$ are the longitudinal and lateral tire forces of the left wheel; $F_{cx'}$ and $F_{cy'}$ are the longitudinal and lateral forces exerted on C by the castor; $F_{ex'}$ and $F_{ey'}$ are the longitudinal and lateral forces exerted on E by the tool (e.g., a robot arm); b, c, d, and e are distances; and τ_e is the moment exerted by the tool.

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

In Fig. 1(a), the dynamic model of the mobile robot is given by

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\psi}\\ \dot{u}\\ \dot{\omega}\end{bmatrix}=\begin{bmatrix}u\cos\psi-a\omega\sin\psi\\ u\sin\psi+a\omega\cos\psi\\ \omega\\ \frac{\theta_3}{\theta_1}\omega^2-\frac{\theta_4}{\theta_1}u\\ -\frac{\theta_5}{\theta_2}u\omega-\frac{\theta_6}{\theta_2}\omega\end{bmatrix}+\begin{bmatrix}0&0\\0&0\\0&0\\ \frac{1}{\theta_1}&0\\ 0&\frac{1}{\theta_2}\end{bmatrix}\begin{bmatrix}uc\\ \omega c\end{bmatrix}+\begin{bmatrix}\delta_x\\ \delta_y\\ 0\\ \delta_u\\ \delta_\omega\end{bmatrix}. \tag{1}$$

The identified parameters θ of the dynamic model (validated in De la Cruz and Carelli[10]) are

$$\theta_1=\left(\frac{R_a}{k_a}(mR_tr+2I_e)+2rk_{DT}\right)/(2rk_{PT})=0.24089;$$

$$\theta_2=\left(\frac{R_a}{k_a}\left(I_ed^2+2R_tr(I_z+mb^2)\right)+2rdk_{DR}\right)\Big/(2rdk_{PR})$$
$$=0.2424;$$

$$\theta_3=\frac{R_a}{k_a}mbR_t/(2k_{PT})=-0.00093603;$$

$$\bar{\theta}_4^0=\frac{R_a}{k_a}\left(\frac{k_ak_b}{R_a}+B_e\right)\Big/(rk_{PT})+1=0.99629;$$

$$\theta_5=\frac{R_a}{k_a}mbR_t/(dk_{PR})=-0.0037256;$$

$$\theta_6=\frac{R_a}{k_a}\left(\frac{k_ak_b}{R_a}+B_e\right)d\Big/(2rk_{PR})+1=1.0915,$$

where m is the robot mass; r is the right and left wheel radius; k_b is equal to the voltage constant multiplied by the gear ratio; R_a is the electric resistance constant; k_a is the torque constant multiplied by the gear ratio; k_{PR}, k_{PT}, and k_{DT} are positive constants; I_e and B_e are the moment of inertia and the viscous friction coefficient of the combined motor rotor, gearbox, and wheel; and R_t is the nominal radius of the tire.[10]

The elements of the uncertainty vector $\boldsymbol{\delta}$ related to the mobile robot are $\boldsymbol{\delta}=[\,\delta_x\ \delta_x\ 0\ \delta_u\ \delta_\omega\,]^T$, where δ_x and δ_y depend on velocities due to wheels' slide and robot orientation; δ_u and δ_ω depend on mechanic parameters of the robot, such as mass, inertial moment, wheel diameter, engine and servo-controllers parameters, forces on the wheels, etc. All these parameters are considered as disturbances.

Remark 1. If wheels' slides, forces, and torques exerted by the tool on the castor wheel are of no significant value, the uncertainties vector will not be considered. We assume that forces and torques exerted by the tool are not measured by a sensor.

In general, most market-available robots have low-level proportional-integral-derivative (PID) velocity controllers to track input reference velocities and do not allow the motor voltage to be driven directly. Therefore, it is useful to express the mobile-robots model in a suitable way by considering rotational and translational reference velocities as control signals.

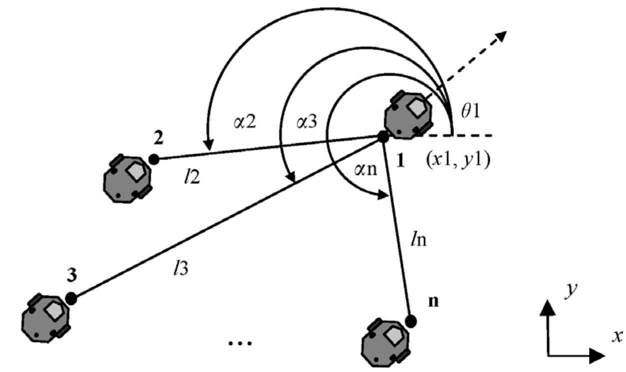

Fig. 2. Formation scheme for n robots.

3. Kinematic Model of the Formation

The kinematic model of the formation describes the time-variation of the multi-robot system. There are several ways to define the formation states. The election of the formation states depends on the application or the task to be performed (e.g., whether one wants the leader of the formation to follow a desired trajectory or the whole group needs to track this trajectory).

3.1. Formation states

Let $\mathbf{z}$ be a vector, which contains a set of formation variables. These variables describe the aspect, position, and orientation of a formation or multi-robot system (e.g., the relative positions between two or more robots, orientation of the formation, etc.). These formation variables and their derivatives will be named as *formation states.*[10] The vector $\mathbf{z}$ could be defined by numerous forms. In Fig. 2 the vector $\mathbf{z}$ has the following components $\{x1, y1, l2, \alpha 2, l3, \alpha 3, \ldots, l\mathrm{n}, \alpha n\}$, where n is the number of robots of the multi-robot system; $\{x1, y1\}$ determine the formation position; $\{l2, l3, \ldots, l\mathrm{n}\}$ and $\{\alpha 2, \alpha 3, \ldots, \alpha\mathrm{n}\}$ determine the aspect of the formation and the relative position of other robots in regard to the leader respectively. Observe that the change of $\mathbf{z}$ to Cartesian coordinates is immediate.

Furthermore, let $\boldsymbol{\xi}=[x1y1\ldots xn\,yn]^T$ be the vector whose components are the robot's positions (Cartesian coordinates). The components of $\boldsymbol{\xi}$ are also formation variables; if we define a state vector by using these variables, this vector will name trivial vector of the formation states. Furthermore, let $\dot{\boldsymbol{\xi}}=[\,\dot{x}1\ \dot{y}1\ \ldots\ \dot{x}\mathrm{n}\ \dot{y}\mathrm{n}\,]^T$ be the vector whose components are the derivatives of $\boldsymbol{\xi}$.

Example. For a triangle-shap formation, we want to control the position of the formation leader, the steering of the formation and its aspect. The chosen formation states are shown in Fig. 3.

The map Φ can be defined as follows:

$$\mathbf{z}=\begin{bmatrix}x1 & y1 & \alpha 2 & l2 & \alpha 3 & l3\end{bmatrix}^T=\Phi(\boldsymbol{\xi}),$$
$$\boldsymbol{\xi}=\begin{bmatrix}x1 & y1 & x2 & y2 & x3 & x3\end{bmatrix}^T. \tag{2}$$

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

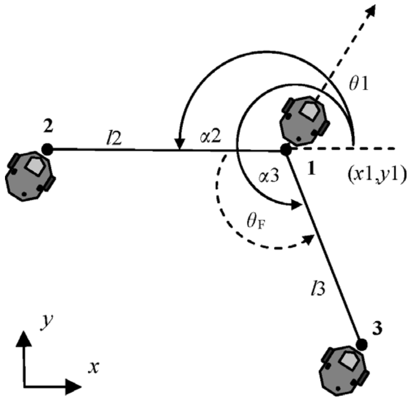

Fig. 3. Formation scheme for three robots.

In (2) $x1$ and $y1$ are the Cartesian coordinates of the leader robot, and the distance ln and the angle αn are defined by

$$ln = \sqrt{(xn - x1)^2 + (yn - y1)^2}, \quad \text{n} = 2, 3, \qquad (3)$$

$$\alpha_{1\text{n}} = f_{\zeta\text{n}}(x\text{n} - x1, y\text{n} - y1), \quad \text{n} = 2, 3, \qquad (4)$$

where, the function $f_{\zeta\text{n}}(x, y)$ allows computing the angle formed by vector $[x\text{n} - x1, y\text{n} - y1]$ with respect to x-axis; this angle is in the range $(\zeta\text{n}, 2\pi + \zeta\text{n})$. The parameter ζn defines the range of the function $f_{\zeta\text{n}}$.

3.2. *Kinematic model of the formation*

The *first-order kinematic model*[10] of the multi-robot system is given by (5)

$$\dot{\boldsymbol{\xi}} = \boldsymbol{\eta} + \boldsymbol{\rho}, \qquad (5)$$

where

$$\boldsymbol{\eta} = \begin{bmatrix} \dot{x}1\text{ref} & \dot{y}1\text{ref} & \dot{x}2\text{ref} & \dot{y}2\text{ref} & \cdots & \dot{x}\text{nref} & \dot{y}\text{nref} \end{bmatrix}^T \qquad (6)$$

is the input of the formation system, with the components of the reference velocities for each robot of the team; $\boldsymbol{\xi}$ is the trivial vector of the formation states; and $\boldsymbol{\rho}$ is a disturbance vector, which contains the differences between the real and reference components of the velocity for each robot.

3.3. *Transformation of coordinates*

To express the model of the multi-robot system by means of the vector of the formation states, we define a smooth map Φ from ξ to $\mathbf{z}$, so that the inverse Φ^{-1} exists and be smooth. Then we define a diffeomorphism[1] Φ of ξ. This diffeomorphism is expressed as a function of Φ.

$$\mathbf{z} = \Phi(\boldsymbol{\xi}), \qquad (7)$$

where, $\mathbf{z}$ is the formation states vector. The map Φ defines a diffeomorphism because the map Φ^{-1} is smooth,

$$\boldsymbol{\xi} = \Phi^{-1}(\mathbf{z}). \qquad (8)$$

If the previous equation is derived, then

$$\dot{\boldsymbol{\xi}} = \mathbf{J}(z)\dot{\mathbf{z}}, \qquad (9)$$

where the Jacobian matrix is defined by

$$\mathbf{J}(z) = \frac{\partial \Phi^{-1}(\mathbf{z})}{\partial \mathbf{z}}. \qquad (10)$$

Replacing the kinematic model (5) into (9) we obtain the kinematic model, which depends on the new coordinate

$$\dot{\mathbf{z}} = \mathbf{J}(z)^{-1}\boldsymbol{\eta} + \mathbf{J}(z)^{-1}\boldsymbol{\rho}, \qquad (11)$$

where, $\mathbf{J}(z)^{-1}$ is the inverse Jacobian matrix, which is a 2n-by-2n matrix (where n is the number of robots of the multi-robot system).

For instance, from the previous example, the inverse map Φ^{-1} is computed as follows:

$$\boldsymbol{\xi} = \begin{bmatrix} x1 \\ y1 \\ x2 \\ y2 \\ x3 \\ y3 \end{bmatrix} = \Phi^{-1}(\mathbf{z}) = \begin{bmatrix} x1 \\ y1 \\ x1 + l2\cos(\alpha 2) \\ y1 + l2\sin(\alpha 2) \\ x1 + l3\cos(\alpha 3) \\ y1 + l3\sin(\alpha 3) \end{bmatrix}. \qquad (12)$$

Then, from (8), (10), and (12), we can express the Jacobian $\mathbf{J}(z)$ as follows:

$$\mathbf{J}(z) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -l2\sin\alpha 2 & \cos\alpha 2 & 0 & 0 \\ 0 & 1 & l2\cos\alpha 2 & \sin\alpha 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & -l3\sin\alpha 3 & \cos\alpha 3 \\ 0 & 1 & 0 & 0 & l3\cos\alpha 3 & \sin\alpha 3 \end{bmatrix}. \qquad (13)$$

Remark 2. Observe that the Jacobian matrix $\boldsymbol{J}(z)$ is a square nonsingular matrix 2n-by-2n, and its inverse always exists. In addition, the determinant of the Jacobian matrix $\mathbf{J}(z)$ is given by the product of the distances between the robots of the formation, that is, $\det(\mathbf{J}(z)) = l2l3\cdots ln$.

4. Formation Controller Design

The controllers proposed in this work are based on linear algebra theory and numerical methods (Scaglia, *et al.* 2008a).[32] By knowledge of the desired state at the next sample time, it is possible to compute the necessary control actions so that the mobile robot tracks the reference trajectory with a good performance. We assume that the mobile-robots formation is moving on a horizontal plane without slip.

[1] A diffeomorphism is a kind of isomorphism (kind of mapping between objects, which shows a relationship between two properties or operations) of smooth manifolds. It is an invertible function that maps one differentiable manifold (abstract mathematical space in which every point has a neighborhood, which resembles Euclidean space, but in which the global structure may be more complicated) to another such that both function and its inverse are smooth.

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

First, we will present the design of the dynamic controller for a mobile robot. This controller receives the velocity references (desired control signals) to track a pre-established trajectory in a precise manner. The velocity references are provided by the formation controller, which allows the multi-robot system achieving the desired formation states with a minimum error.

4.1. Dynamic model-based controller by using linear algebra

The following set of equations is obtained through Euler approximations[2] of the dynamic model of the mobile robot (1) (Rosales, *et al.* 2009),

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \psi_{k+1} \\ u_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \psi_k \\ u_k \\ \omega_k \end{bmatrix} + To \left\{ \begin{bmatrix} u_k \cos\psi_k - a\omega_k \sin\psi_k \\ u_k \sin\psi_k + a\omega_k \cos\psi_k \\ \omega_k \\ \frac{\theta_3}{\theta_1}\omega_k^2 - \frac{\theta_4}{\theta_1}u_k \\ -\frac{\theta_5}{\theta_2}u_k\omega_k - \frac{\theta_6}{\theta_2}\omega_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix} \begin{bmatrix} uc_k \\ \omega c_k \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix} \right\}, \qquad (14)$$

where, values of x at the discrete time $t = kTo$ will be denoted as x_k, To is the sample time, and $k = 0, 1, 2, \ldots$ Afterwards, the state vector $\mathbf{x}_{k+1}$ is replaced by the desired state vector (a desired trajectory can be obtained through a planning algorithm or a generator of precomputed paths).

$$\mathbf{x}d_{k+1} = \left[xd_{k+1} \; yd_{k+1} \; \psi d_{k+1} \; ud_{k+1} \; \omega d_{k+1} \right]^T.$$

Then, from (14), we form the following system of linear equations

$$\mathbf{A}\boldsymbol{\mu}_k = \mathbf{b}, \qquad (15)$$

where

$$\boldsymbol{\mu}_k = \left[uc_k \quad \omega c_k \right]^T, \qquad (16)$$

$$\mathbf{b} = \begin{bmatrix} xd_{k+1} - x_k \\ yd_{k+1} - y_k \\ \psi d_{k+1} - \psi_k \\ ud_{k+1} - u_k \\ \omega d_{k+1} - \omega_k \end{bmatrix} - To \begin{bmatrix} u_k \cos\psi_k - a\omega_k \sin\psi_k \\ u_k \sin\psi_k + a\omega_k \cos\psi_k \\ \omega_k \\ \frac{\theta_3}{\theta_1}\omega_k^2 - \frac{\theta_4}{\theta_1}u_k \\ -\frac{\theta_5}{\theta_2}u_k\omega_k - \frac{\theta_6}{\theta_2}\omega_k \end{bmatrix} - To \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix}, \qquad (17)$$

$$\mathbf{A} = To \begin{bmatrix} 0 & 0 & 0 & \frac{1}{\theta_1} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\theta_2} \end{bmatrix}^T. \qquad (18)$$

From (15), which is a set of five equations with two unknown variables, and by using normal equations[3] ($\mathbf{A}^T\mathbf{A}\boldsymbol{\mu}_k = \mathbf{A}^T\mathbf{b}$),[35] we find the optimal solution (according to minimal squares[4]) for $\boldsymbol{\mu}_k$,

$$\begin{bmatrix} uc_k \\ \omega c_k \end{bmatrix} = \begin{bmatrix} k_u \dfrac{\theta_1 (ud_{k+1} - u_k) - To\left(\theta_3\omega_k^2 - \theta_4 u_k + \theta_1\delta_u\right)}{To} \\ k_\omega \dfrac{\theta_2 (\omega d_{k+1} - \omega_k) - To\left(-\theta_5 u_k\omega_k - \theta_6\omega_k + \theta_2\delta_\omega\right)}{To} \end{bmatrix}, \qquad (19)$$

where, k_u and k_ω are positive constants that allow us adjusting the performance of the proposed control system; these constants satisfy $0 < (k_u, k_\omega) < 1$, allowing to reduce the variations in state variables. Furthermore, ud_{k+1} and ωd_{k+1} are the linear and rotational desired velocities, respectively.

Remark 3. A set of both simulation and experimentation tests were developed with $k_u = k_\omega = 1$ in (19). During these evaluations, it can be noticed that the mobile robot follows the desired trajectory, but in an oscillatory way (Rosales, *et al.* 2009). In order to correct this undesired behavior, the control actions can be calculated by the minimization of a quadratic index, in which not only the tracking error but also the square of state variable derivatives has been considered, as seen in (20). Hence, the minimization of the tracking error as well as that of the variation of the state variables are considered,

$$\Gamma = k_1^2\left[(xd_{k+1} - x_{k+1})^2 + (yd_{k+1} - y_{k+1})^2\right] + k_2^2(\psi d_{k+1} - \psi_{k+1})^2 + k_3^2(ud_{k+1} - u_{k+1})^2 + k_4^2(\omega d_{k+1} - \omega_{k+1})^2 + k_5^2 \underbrace{\left(\dot{x}_k^2 + \dot{y}_k^2\right)}_{uc_k^2} + k_6^2 \underbrace{\dot{\psi}_k}_{\omega c_k^2}, \qquad (20)$$

[2] Euler approximations are a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value.

[3] Given an overdetermined matrix equation $\mathbf{Ax} = \mathbf{b}$, the normal equation is that which minimizes the sum of the square differences between left and right sides of $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$. Here, $\mathbf{A}^T\mathbf{A}$ is a normal matrix, that is, $\mathbf{A}^T\mathbf{A} - \mathbf{A}\mathbf{A}^T = \mathbf{0}$.

[4] The least squares solution to an inconsistent system $\mathbf{Ax} = \mathbf{b}$ satisfies $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$. If the columns of $\mathbf{A}$ are linearly independent, $\mathbf{A}^T\mathbf{A}$ is invertible and $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$.

where k_1, k_2, k_3, k_4, k_5, and k_6 are constants; and x_{k+1}, y_{k+1}, ψ_{k+1}, u_{k+1}, and ω_{k+1} are given by (14). We are looking for the derivative of the proposed index (20) with respect to the control actions (16) to make a minimization. Next, working on the previous index the following expressions can be reached:

$$\frac{\partial\,\Gamma}{\partial\,uc_k} = 2\frac{To}{\theta_1}k_3^2\left[ud_{k+1} - u_k - To\left(\frac{\theta_3}{\theta_1}\omega_k^2 - \frac{\theta_4}{\theta_1}u_k + \frac{1}{\theta_1}uc_k + \delta_u\right)\right] + 2k_5^2 uc_k = 0, \tag{21}$$

$$\frac{\partial\,\Gamma}{\partial\,\omega c_k} = 2\frac{To}{\theta_2}k_4^2\left[\omega d_{k+1} - \omega_k + To\left(\frac{\theta_5}{\theta_2}u_k\omega_k + \frac{\theta_6}{\theta_2}\omega_k - \frac{1}{\theta_2}\omega c_k - \delta_\omega\right)\right] + 2k_6^2\omega c_k = 0. \tag{22}$$

From (21) and (22), the expressions for the control actions can be obtained,

$$\begin{bmatrix} uc_k \\ \omega c_k \end{bmatrix} = \begin{bmatrix} \underbrace{\left(\dfrac{k_3^2}{k_3^2 + \frac{\theta_1^2}{To^2}k_5^2}\right)}_{k_u} \times \dfrac{\theta_1(ud_{k+1} - u_k) - To\left(\theta_3\omega_k^2 - \theta_4 u_k + \theta_1\delta_u\right)}{To} \\ \underbrace{\left(\dfrac{k_4^2}{k_4^2 + \frac{\theta_2^2}{To^2}k_6^2}\right)}_{k_\omega} \times \dfrac{\theta_2(\omega d_{k+1} - \omega_k) - To\left(-\theta_5 u_k\omega_k - \theta_6\omega_k + \theta_2\delta_\omega\right)}{To} \end{bmatrix}. \tag{23}$$

If (19) and (23) are compared, then it can be seen that, to minimize the state variables variations, the constant values of k_u and k_ω should be chosen less than 1.

Now, the objective is to find ud_{k+1} and ωd_{k+1} for each of the robots so that their tracking error is minimal. For this purpose, a linear algebra-based formation controller will be designed.

4.2. Formation controller by using linear algebra

Once more, by using the Euler approximations, the kinematic model of the formation (11) is discretized

$$\mathbf{z}_{k+1} = \mathbf{z}_k + To\mathbf{J}(z_k)^{-1}\{\boldsymbol{\eta}_k + \boldsymbol{\rho}_k\}, \tag{24}$$

so that, from (6), (24), and by using $\mathbf{z}_k = \left[x1_k\ y1_k\ \alpha 2_k\ l2_k \cdots \alpha 3_k\ l3_k\right]^T$, the kinematic first-order model of the formation can be expressed as follows:

$$\begin{bmatrix} x1_{k+1} \\ y1_{k+1} \\ \alpha 2_{k+1} \\ l2_{k+1} \\ \vdots \\ \alpha n_{k+1} \\ ln_{k+1} \end{bmatrix} = \begin{bmatrix} x1_k \\ y1_k \\ \alpha 2_k \\ l2_k \\ \vdots \\ \alpha n_k \\ ln_k \end{bmatrix} + To\mathbf{J}(z_k)^{-1}\begin{bmatrix} \dot{x}1\mathrm{ref}_k \\ \dot{y}1\mathrm{ref}_k \\ \dot{x}2\mathrm{ref}_k \\ \dot{y}1\mathrm{ref}_k \\ \vdots \\ \dot{x}n\mathrm{ref}_k \\ \dot{y}n\mathrm{ref}_k \end{bmatrix} + To\mathbf{J}(z_k)^{-1}\begin{bmatrix} \rho_{\dot{x}1} \\ \rho_{\dot{y}1} \\ \rho_{\dot{x}2} \\ \rho_{\dot{y}2} \\ \vdots \\ \rho_{\dot{x}n} \\ \rho_{\dot{y}n} \end{bmatrix}. \tag{25}$$

Next, the formation states vector ($\mathbf{z}_{k+1}$) is replaced by the desired one ($\mathbf{z}d_{k+1}$).

$$\mathbf{z}d_{k+1} = \left[x1d_{k+1}\ y1d_{k+1}\ \alpha 2d_{k+1}\ l2d_{k+1} \cdots \alpha nd_{k+1}\ lnd_{k+1}\right]^T.$$

From (24) and (25), we form the following system of linear equations:

$$\mathbf{D}\boldsymbol{\eta}_k = \mathbf{e}, \tag{26}$$

where

$$\boldsymbol{\eta}_k = \left[\dot{x}1\mathrm{ref}_k\ \ \dot{y}1\mathrm{ref}_k\ \ \dot{x}2\mathrm{ref}_k\ \ \dot{y}2\mathrm{ref}_k\ \cdots\ \dot{x}n\mathrm{ref}_k\ \ \dot{y}n\mathrm{ref}_k\right]^T, \tag{27}$$

$$\mathbf{e} = \begin{bmatrix} x1d_{k+1} \\ y1d_{k+1} \\ \alpha 2d_{k+1} \\ l2d_{k+1} \\ \vdots \\ \alpha nd_{k+1} \\ lnd_{k+1} \end{bmatrix} - \begin{bmatrix} x1_k \\ y1_k \\ \alpha 2_k \\ l2_k \\ \vdots \\ \alpha n_k \\ ln_k \end{bmatrix} - To\mathbf{J}\left(z_k\right)^{-1}\begin{bmatrix} \rho_{\dot{x}1} \\ \rho_{\dot{y}1} \\ \rho_{\dot{x}2} \\ \rho_{\dot{y}2} \\ \vdots \\ \rho_{\dot{x}n} \\ \rho_{\dot{y}n} \end{bmatrix}, \tag{28}$$

$$\mathbf{D} = To\mathbf{J}(z_k)^{-1}. \tag{29}$$

From (26), which is a set of 2n equations with 2n unknown variables, we find the exact solution[35] for $\boldsymbol{\eta}_k = \mathbf{D}^{-1}\mathbf{e}$,

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

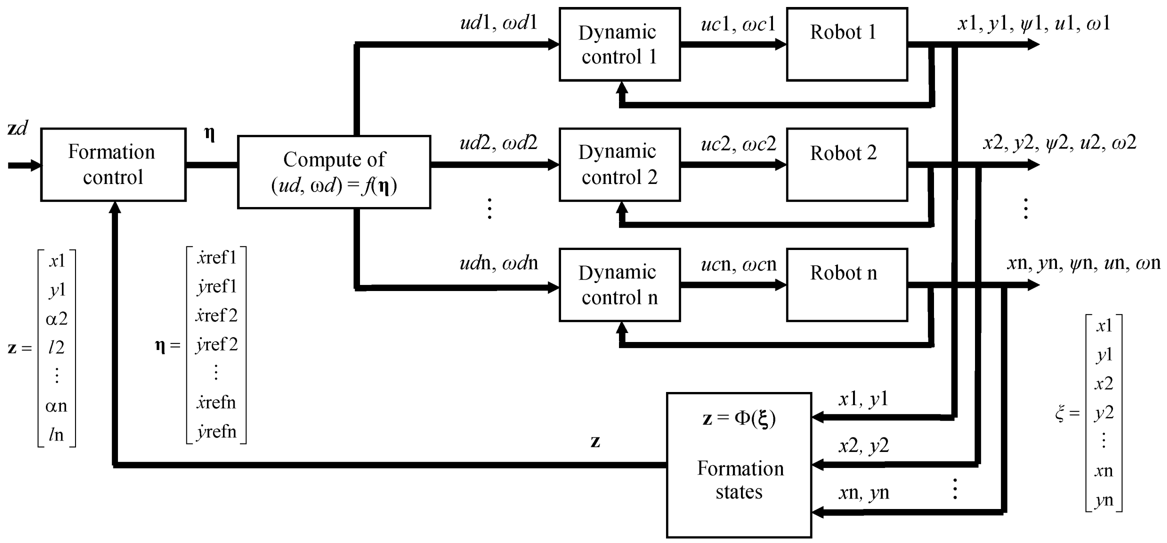

Fig. 4. Blocks diagram for the formation control.

that is,

$$\begin{bmatrix} \dot{x}1\text{ref}_k \\ \dot{y}1\text{ref}_k \\ \dot{x}2\text{ref}_k \\ \dot{y}1\text{ref}_k \\ \vdots \\ \dot{x}\text{nref}_k \\ \dot{y}\text{nref}_k \end{bmatrix} = \frac{1}{To} \begin{bmatrix} k_{1x}\left\{x1d_{k+1} - x1_k - To\rho_{\dot{x}1}\right\} \\ k_{1y}\left\{y1d_{k+1} - y1_k - To\rho_{\dot{y}1}\right\} \\ k_{2x}\left\{x1d_{k+1} - x1_k - l2(\alpha 2d_{k+1} - \alpha 2_k)\sin(\alpha 2) + (l2d_{k+1} - l2_k)\cos(\alpha 2) - To\rho_{\dot{x}2}\right\} \\ k_{2y}\left\{y1d_{k+1} - y1_k + l2(\alpha 2d_{k+1} - \alpha 2_k)\cos(\alpha 2) + (l2d_{k+1} - l2_k)\sin(\alpha 2) - To\rho_{\dot{y}2}\right\} \\ \vdots \\ k_{\text{n}x}\left\{x1d_{k+1} - x1_k - l\text{n}(\alpha\text{n}d_{k+1} - \alpha\text{n}_k)\sin(\alpha\text{n}) + (l\text{n}d_{k+1} - l\text{n}_k)\cos(\alpha\text{n}) - To\rho_{\dot{x}\text{n}}\right\} \\ k_{\text{n}y}\left\{y1d_{k+1} - y1_k + l\text{n}(\alpha\text{n}d_{k+1} - \alpha\text{n}_k)\cos(\alpha\text{n}) + (l\text{n}d_{k+1} - l\text{n}_k)\sin(\alpha\text{n}) - To\rho_{\dot{y}\text{n}}\right\} \end{bmatrix}, \tag{30}$$

where $k_{\text{i}x}$ and $k_{\text{i}y}$ with $\text{i} = 1, 2, 3, \ldots, \text{n}$ are positive constants that allow us adjusting the performance of the proposed control system; these constants satisfy $0 < (k_{\text{i}x}, k_{\text{i}y}) < 1$, allowing to reduce the variations in the formation variables (see Remark 3). From (30), we can obtain the linear and rotational desired velocities for (23). We can compute ud_{k+1} and ωd_{k+1} for each of the robots by means of a diffeomorphism Ω of $(\dot{x}\text{ref}_k, \dot{y}\text{ref}_k)$. This diffeomorphism is expressed as a function of Ω,

$$\begin{bmatrix} ud1_{k+1} \\ \omega d1_{k+1} \\ ud2_{k+1} \\ \omega d2_{k+1} \\ \vdots \\ ud\text{n}_{k+1} \\ \omega d\text{n}_{k+1} \end{bmatrix} = \Omega\left(\dot{x}\text{ref}_k, \dot{y}\text{ref}_k\right) = \begin{bmatrix} \sqrt{(\dot{x}1\text{ref}_k)^2 + (\dot{y}1\text{ref}_k)^2} \\ \arctan\left(\dot{y}1\text{ref}_k, \dot{x}1\text{ref}_k\right) \\ \sqrt{(\dot{x}2\text{ref}_k)^2 + (\dot{y}2\text{ref}_k)^2} \\ \arctan\left(\dot{y}2\text{ref}_k, \dot{x}2\text{ref}_k\right) \\ \vdots \\ \sqrt{(\dot{x}\text{nref}_k)^2 + (\dot{y}\text{nref}_k)^2} \\ \arctan\left(\dot{y}\text{nref}_k, \dot{x}\text{nref}_k\right) \end{bmatrix}. \tag{31}$$

We propose using these values like the desired ones at instant $k + 1$ for the linear and rotational desired velocities.

Remark 4. The use of numerical methods to compute the system's evolution is based mainly on the possibility to determine the system state at instant $k + 1$, if the state is known at instant k (Markov property[5]). So, a variable at instant $k + 1$ can be substituted for the desired one and subsequently computing the necessary control action to make the states of the system move from its current value to the desired one.

Figure 4 shows a block diagram for the formation control, where the location of the dynamic controllers for each robot and the location of the formation controller for the multi-robot system can be seen.

Remark 5. The formation control law (30) is *scalable*, because it does not depend on the number of robots. Scalability means that the control law is able to expand easily to any number of robots.[3] Therefore, from (13) we can extend

[5] Having the Markov property means that, given the present state, the future states are independent of the past ones. In other words, the description of the present state fully captures all the information that could influence the future evolution of the process.

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

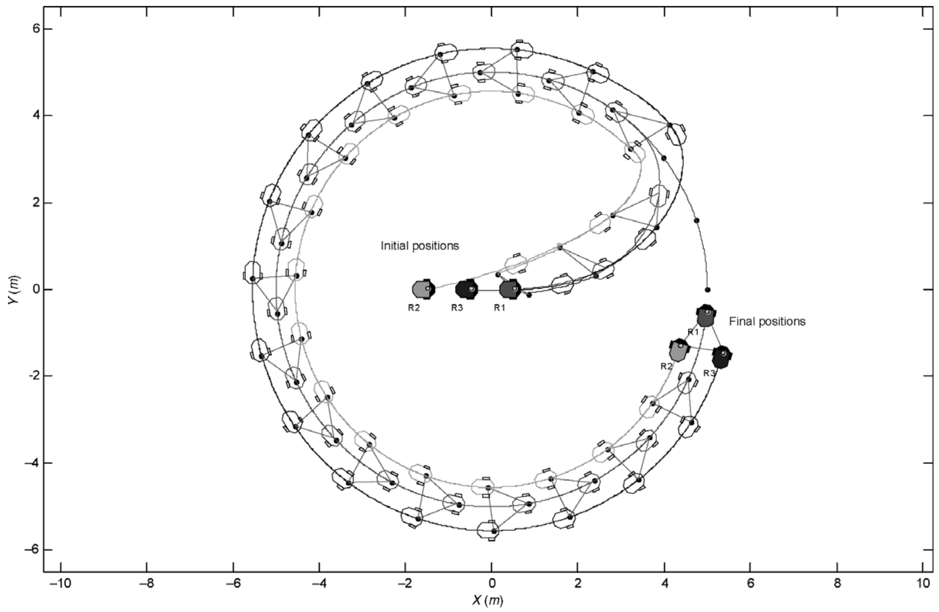

Fig. 5. Trajectories of the multi-robot system with a circle reference.

the Jacobian matrix for n robots,

$$\mathbf{J}(z)=\begin{bmatrix}\mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2}\\ \mathbf{I}_{2\times2} & \mathbf{J}_2 & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2}\\ \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{J}_3 & \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2}\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} & \mathbf{J}_\mathrm{n}\end{bmatrix}, \tag{32}$$

where $\mathbf{I}_{2\times2}$ is the identity matrix of size 2; $\mathbf{0}_{2\times2}$ is a null square-matrix of size 2; and $\boldsymbol{J}_\mathrm{n}$ is a nonsingular square-matrix defined by

$$\mathbf{J}_\mathrm{n}=\begin{bmatrix}-l\mathrm{n}\sin(\alpha\mathrm{n}) & \cos(\alpha\mathrm{n})\\ l\mathrm{n}\cos(\alpha\mathrm{n}) & \sin(\alpha\mathrm{n})\end{bmatrix}. \tag{33}$$

Remark 6. The proposed map Φ generates a *decentralized* control law for the formation. The control law (30) only depends on the states of the n-th robot and the states of the leader.

Remark 7. The proposed controller is able to track a moving or continuously varying desired formation.

Remark 8. As $\Phi^{-1}(.)$ is a continuous function $\Phi^{-1}(\mathbf{z}) \to \Phi^{-1}(\mathbf{z}d)$ with $\mathbf{z} \to \mathbf{z}d$, from (8), (9), and (10) $\boldsymbol{\xi} \to \boldsymbol{\xi}d$ with $\mathbf{z} \to \mathbf{z}d$, where $\boldsymbol{\xi}d = \Phi^{-1}(\mathbf{z}d)$.

5. Simulations and Experimental Results

Simulations and experimental results demonstrate the versatility of the control architecture for the multi-robot system. These results are presented in the next figures. The multi-robot system is formed by the robots R1, R2, and R3. The leader robot R1 is scheduled to follow different trajectories and robots R2 and R3 (followers) must maintain the requested formation. Initially, the robots are not in formation, they start with a random location and orientation. The sample time used in all simulations was $To = 0.1$ s. The reference trajectory provides the two first desired formation states $\{x1d, y1d\}$. The other desired formation parameters $\{l2; l3; \theta_F\}$ are supplied according to the necessary task or mission of the group. The required formation shape will be represented by dashed gray lines both in simulation and experimental results. θ_F is an angle that allows computing $\alpha2$ and $\alpha3$ (see. Fig. 3) by using $\psi1$ (the orientation angle of the leader robot R1), that is,

$$\alpha2=\psi1+\pi-\frac{\theta_\mathrm{F}}{2}\ \text{and}\ \alpha3=\psi1+\pi+\frac{\theta_\mathrm{F}}{2}.$$

5.1. Simulation results

The computer simulations were generated by MobileSim software for debugging and experimentation with ARIA or other software that supports mobile-robots platforms (http://robots.mobilerobots.com/). The simulated robots are based on the physical mobile robots; therefore the same algorithms were used both in real experiments and simulations. All the graphical results presented in this work were generated by using Matlab software.

The first reference trajectory was a circle of 5000 mm radius with a linear speed $u\mathrm{ref} = 262.5\,\mathrm{mm/s}$ and a rotational speed $\omega\mathrm{ref} = 25.1°/\mathrm{s}$, and initial conditions for the robots were as follows: R1_0: $\{x1_0 = 500\,\mathrm{mm};\ y1_0 = 0\,\mathrm{mm};\ \psi1_0 = 0°\}$, R$2_0$: $\{x2_0 = -1500\,\mathrm{mm};\ y2_0 = 0\,\mathrm{mm};\ \psi2_0 = 0°\}$, and R$3_0$: $\{x3_0 = -500\,\mathrm{mm};\ y3_0 = 0\,\mathrm{mm};\ \psi3_0 = 0°\}$. Also, the desired formation parameters for this case were $\{l2 = 1000\,\mathrm{mm};\ l3 = 1000\,\mathrm{mm};\ \theta_\mathrm{F} = 60°\}$.

Figure 5 shows that the multi-robot system follows the desired circular-trajectory in a precise manner. At the beginning, the robots start at random positions, and at the same time as the leader reaches the trajectory, the followers arrive at their formation positions. Next, when the formation positions have been fulfilled, the multi-robot system keeps

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

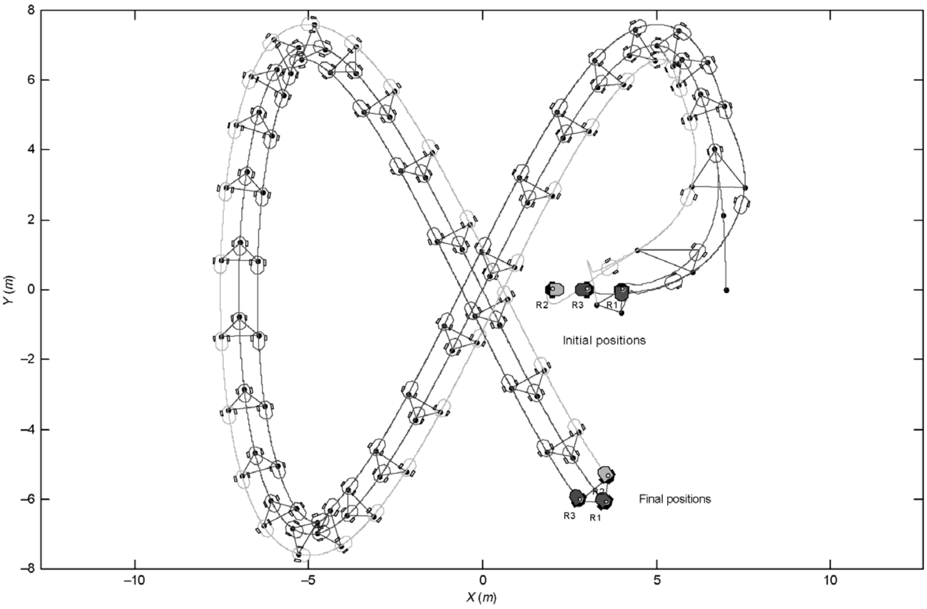

Fig. 6. Trajectories of the multi-robot system with an eight-shaped reference.

the pre-established configuration during the whole test with a minimum error less than 65 mm for x and y.

The second case was carried out with an eight-shaped trajectory of 7000 mm radius with a linear speed $u\text{ref} = 300\,\text{mm/s}$ and a rotational speed $\omega\text{ref} = 24.56°/\text{s}$. The results are presented in Fig. 6. In this test, the initial conditions for the robots were $\text{R}1_0$: $\{x1_0 = 4000\,\text{mm};\ y1_0 = 0\,\text{mm};\ \psi1_0 = 90°\}$, $\text{R}2_0$: $\{x2_0 = 3000\,\text{mm};\ y2_0 = 0\,\text{mm};\ \psi2_0 = 180°\}$, and $\text{R}3_0$: $\{x3_0 = 2000\,\text{mm};\ y3_0 = 0\,\text{mm};\ \psi3_0 = 0°\}$. Moreover, the desired formation parameters in this case were $\{l2 = 750\,\text{mm};\ l3 = 750\,\text{mm};\ \theta_\text{F} = 90°\}$.

In Fig. 6, the performance of the multi-robot system with an eight-shaped trajectory is presented. Like the previous case, the robots start at random positions, but now they have also random headings. When the formation positions have been reached, the multi-robot system keeps the desired structure with a minimum error less than 85 mm for x and y (in the sharp turning parts) despite the demand of curves in the trajectory.

The last case is shown in Fig. 7; in this case a linear trajectory has been used with a linear speed $u\text{ref} = 300\,\text{mm/s}$.

The initial conditions for the robots were $\text{R}1_0$: $\{x1_0 = -1000\,\text{mm};\ y1_0 = -500\,\text{mm};\ \psi1_0 = 0°\}$, $\text{R}2_0$: $\{x2_0 = -2000\,\text{mm};\ y2_0 = 0\,\text{mm};\ \psi2_0 = 135°\}$, and $\text{R}3_0$: $\{x3_0 = -1000\,\text{mm};\ y3_0 = -1500\,\text{mm};\ \psi3_0 = 225°\}$. In addition, in this case the desired formation parameters were changed while the multi-robot system tracks the reference. The initial formation parameters were $\{l2 = 1250\,\text{mm};\ l3 = 1250\,\text{mm};\ \theta_\text{F} = 60°\}$, then, at time $kTo = 20$ s these parameters were changed to $\{l2 = 600\,\text{mm};\ l3 = 1200\,\text{mm};\ \theta_\text{F} = 0°\}$, and finally at time $kTo = 40$ s, the requested formation parameters were $\{l2 = 750\,\text{mm};\ l3 = 750\,\text{mm};\ \theta_\text{F} = 120°\}$.

Formation changes have been tested in the third case. Figure 7 shows that the multi-robot system follows the desired trajectory in a precise manner while the formation is changed. The robots start at random positions and headings, and during the tracking, thrice change their configurations. First, the multi-robot system is required to form a triangle-shape; then, a column-shape is required; and finally, another triangle-shape is again required. The multi-robot system develops all configurations in a precise way with a minimum error less than 70 mm for x and y.

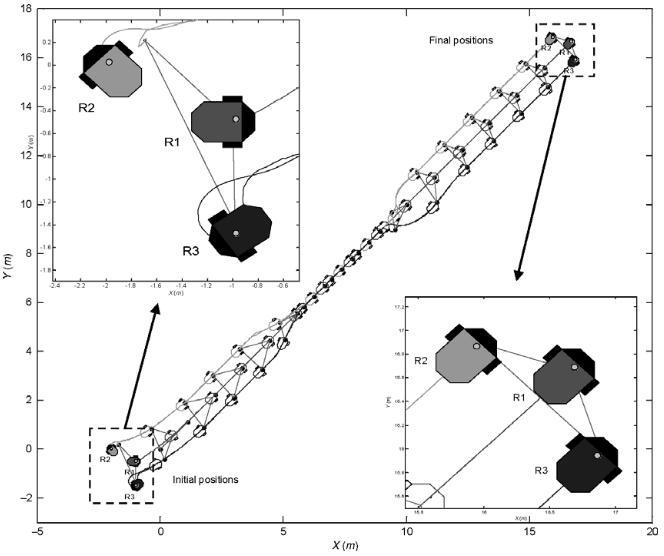

Fig. 7. Trajectories of the multi-robot system with a linear reference.

5.2. *Experimental results*

The proposed methodology has been tested with three PIONEER 3DX mobile robots (see Fig. 8), with approximate 400 mm radius. A sample time $To = 0.1$ s was used and $a = 200$ mm (see Fig. 1). PIONEER mobile robots include an estimation system based on odometry, which adds accumulative errors to the system.[27] From this, data updating through external sensors is necessary. This problem is separated from the strategy of trajectory tracking, which is

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

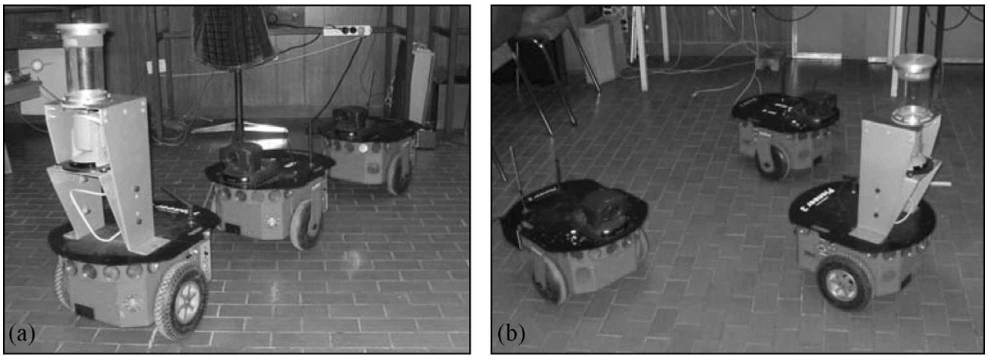

Fig. 8. PIONEER mobile robots and their environment – (a) column-formation; (b) triangle-formation.

not considered in this paper.[25] The robots can communicate with each other through ethernet with TCP/IP protocols. Data acquisition and all necessary algorithms are computed in real-time with the on-board robot computers.

The first experiment was carried out with a linear trajectory with a linear speed $u\text{ref} = 300\,\text{mm/s}$. The initial conditions for the robots were R1_0: $\{x1_0 = 0\,\text{mm}; y1_0 = 0\,\text{mm}; \psi 1_0 = 0°\}$, R$2_0$: $\{x2_0 = -1300\,\text{mm}; y2_0 = 500\,\text{mm}; \psi 2_0 = 90°\}$, and R$3_0$: $\{x3_0 = -800\,\text{mm}; y3_0 = -1100\,\text{mm}; \psi 3_0 = 270°\}$. The formation parameters were $\{l2 = 800\,\text{mm}; l3 = 800\,\text{mm}; \theta_F = 60°\}$. Then, at time $kTo = 17$ s, the desired trajectory stops in $\{x1_{17} = 5000\,\text{mm}; y11_7 = 0\,\text{mm}\}$ and a new orientation was provided to the leader $\psi 11_7 = 45°$.

Figure 9 shows that the multi-robot system follows the desired straightforward-trajectory in a precise manner. At the beginning, the robots start at random positions and headings, and at the same time as the leader reaches the trajectory, the followers arrive at their formation positions. When the formation positions have been fulfilled, the multi-robot system keeps the pre-established configuration with a minimum error. At the end, the formation is positioned with the desired heading of the leader.

The leader motion control gives the robot the capability to get the desired posture in the world coordinate system. The proposed controller allows the follower robots to reach their headings at the desired computed values that will provide good initial heading conditions for future formation missions (see Fig. 10(a)). Taking advantage of the unicycle dynamics, it will be assumed from here the robots can rotate without distorting the formation (allowing to change the "formation heading").

The quadratic formation errors are presented in Fig. 10(b). The simulation starts at time $kTo = 0$ s with a requested triangular formation. Next, at time $kTo = 17$ s, a fixed point is required for the multi-robot system. Therefore, the formation errors are increased during this transition movement, but these errors are reduced when the group reaches the required location. The minimum error was less than 76 mm for x and y.

Figure 11 exhibits the experimental linear and rotational velocity profiles for R1 (dotted line), R2 (solid line), and R3 (dashed line). When the followers try to reach their formation positions, their velocities are increased till the speed limit, then the whole formation keeps the same behavior. At the end, once again, the robots modify their velocities to achieve the final required formation.

The second experiment is presented in Fig. 12. This test was carried out with a circular trajectory with a linear speed $u\text{ref} = 300\,\text{mm/s}$ and a rotational speed $\omega\text{ref} = 24.56°/\text{s}$. The initial conditions for the robots were R1_0: $\{x1_0 = 3500\,\text{mm}; y1_0 = -1500\,\text{mm}; \psi 1_0 = 90°\}$, R$2_0$: $\{x2_0 = 3000\,\text{mm}; y2_0 = -500\,\text{mm}; \psi 2_0 = 90°\}$, and R$3_0$: $\{x3_0 = 4000\,\text{mm}; y3_0 = -500\,\text{mm}; \psi 3_0 = 90°\}$. The initial required formation parameters were $\{l2 = 600\,\text{mm}; l3 = 600\,\text{mm}; \theta_F = 60°\}$. Then, at time $kTo = 19$ s, the desired

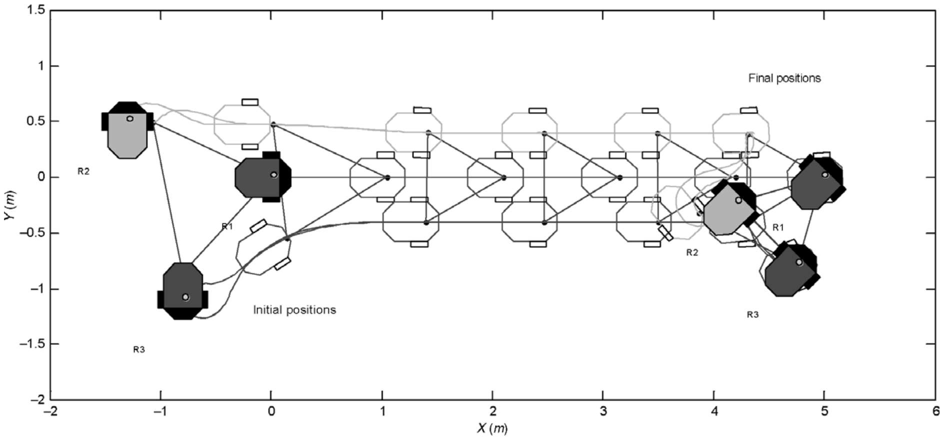

Fig. 9. Experimental trajectories of the multi-robot system with a linear reference. Positioning and heading control.

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

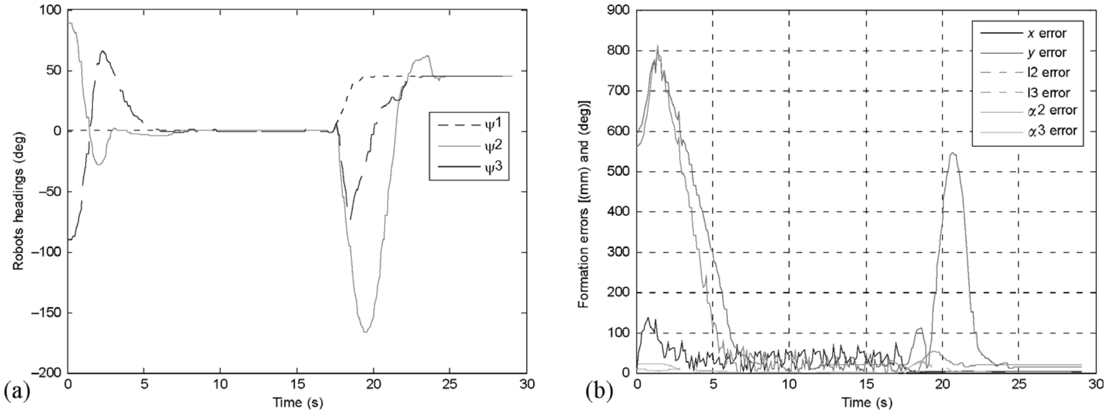

Fig. 10. (a) Headings of robots R1, R2, and R3 for the tracking and positioning control; (b) formation errors for $\{x1,\ y1,\ l2,\ l3,\ \alpha 2,\ \alpha 3\}$ with a linear reference. At time $kTo = 17$(s), a fixed point with a pre-established orientation ($\psi 1_{17} = 45°$) is required for the multi-robot system.

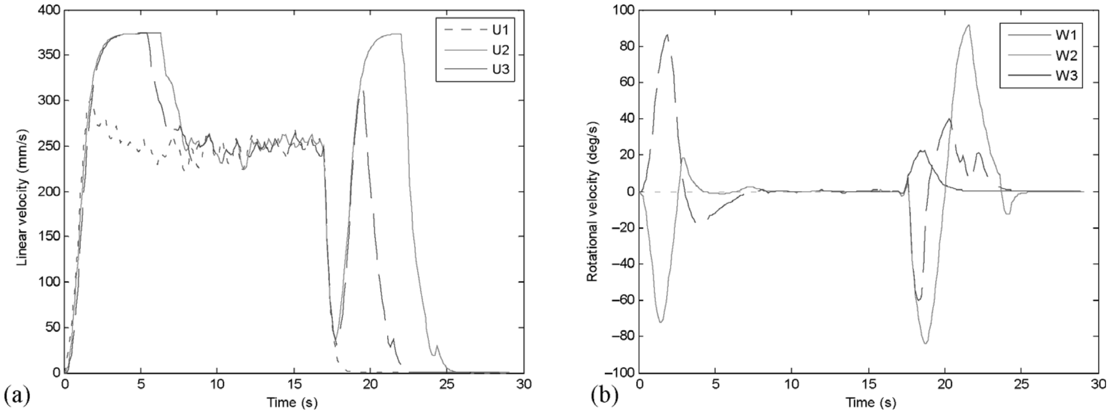

Fig. 11. Experimental velocity profiles for R1, R2, and R3 – (a) linear; (b) rotational.

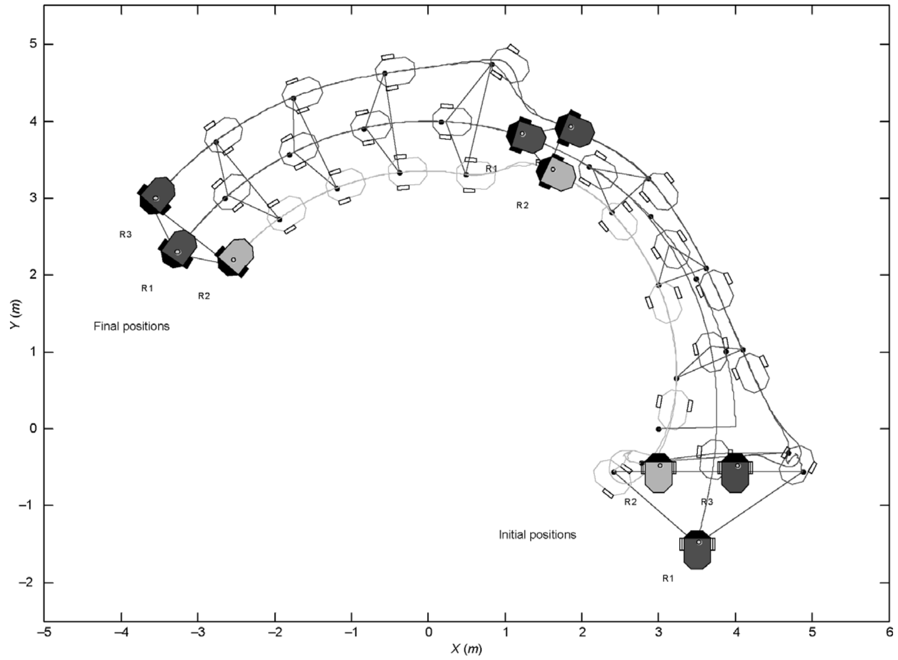

Fig. 12. Experimental trajectories of the multi-robot system with a circular reference. Control of change of formation during the tracking.

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

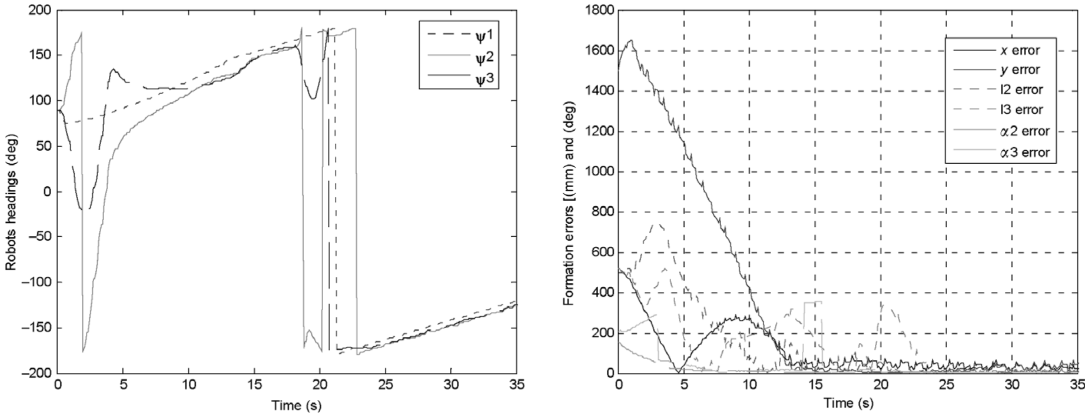

Fig. 13. (a) Headings of robots R1, R2, and R3 for the tracking control and change of formation; (b) formation errors for $\{x1,\ y1,\ l2,\ l3,\ \alpha 2,\ \alpha 3\}$ with a circular reference. At time $kTo = 19$(s), the multi-robot system is required to change the formation.

formation parameters were changed to $\{l2 = 750\,\text{mm};\ l3 = 750\,\text{mm};\ \theta_F = 120°\}$.

A flexible formation can be used in military, rescue, or recognition missions. The proposed strategy allows the team to change its structure during the trajectory tracking, as can be seen in Fig. 12. This second experiment shows that the multi-robot system follows in a precise way the desired circular trajectory. First, the robots start at random positions, and at the same time as the leader reaches the trajectory, the followers arrive at their formation positions. When the formation positions have been fulfilled, the multi-robot system changes its first configuration and forms a different triangle-shaped structure.

Figure 13(a) shows the headings of the mobile robots: R1 (dotted line), R2 (solid line), and R3 (dashed line). The quadratic formation errors are presented in Fig. 13(b). The simulation starts at time $kTo = 0$ s with the first requested triangular formation. Next, at time $kTo = 19$ s, a second triangular formation is required; therefore, the formation errors are increased during this transition movement, but these errors are quickly decreased when the team achieves the new request. The minimum error was less than 53 mm for x and y. Figure 14 exhibits the experimental linear and rotational velocity profiles for R1 (dotted line), R2 (solid line), and R3 (dashed line).

Finally, the last experiment was carried out with a reference at right angle with a linear speed $u\text{ref} = 245\,\text{mm/s}$. It is presented in Fig. 15. The initial conditions for the robots were $R1_0$: $\{x1_0 = -1000\,\text{mm};\ y1_0 = 0\,\text{mm};\ \psi 1_0 = 180°\}$, $R2_0$: $\{x2_0 = -2000\,\text{mm};\ y2_0 = 750\,\text{mm};\ \psi 2_0 = 180°\}$, and $R3_0$: $\{x3_0 = -2000\,\text{mm};\ y3_0 = -750\,\text{mm};\ \psi 3_0 = 180°\}$. The desired formation parameters were $\{l2 = 900\,\text{mm};\ l3 = 900\,\text{mm}; \theta_F = 60°\}$. When the multi-robot system reaches the point (8000 mm, 0 mm), the leader is required to turn its heading at an angle of −90, therefore the followers also do it without losing the formation (shape and size). Once the followers achieve the required headings, the leader continues tracking the trajectory. The leader stops at the point (8000 mm, −5000 mm).

A challenging trajectory to test the performance of the proposed algorithm has been used in this last case. The formation must change swiftly its heading without the loss of its configuration. The multi-robot system develops this test in a precise manner with a minimum error. Figure 16 shows the headings and the experimental linear and rotational velocity profiles of the mobile robots: R1 (dotted line), R2 (solid line), and R3 (dashed line). The heading control is a very important task for a multi-robot system; for instance, starting with a null error formation, the leader develops a

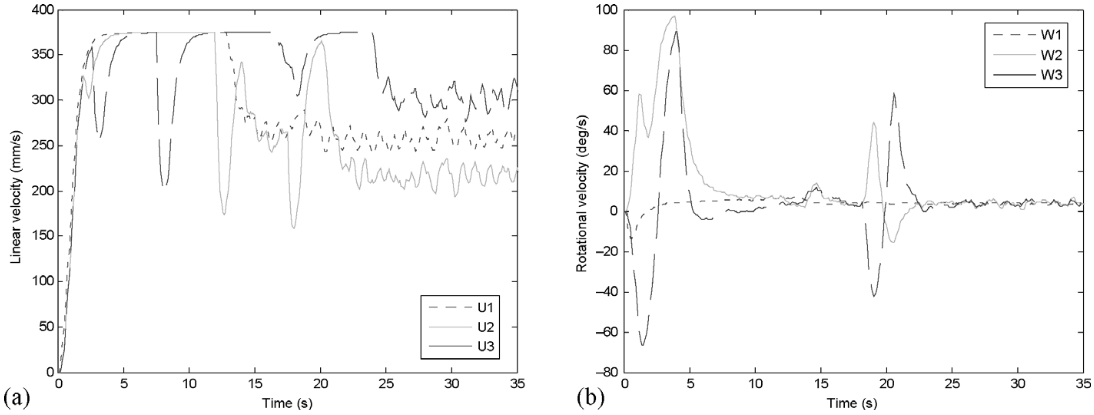

Fig. 14. Experimental velocity profiles for R1, R2, and R3 – (a) linear; (b) rotational.

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

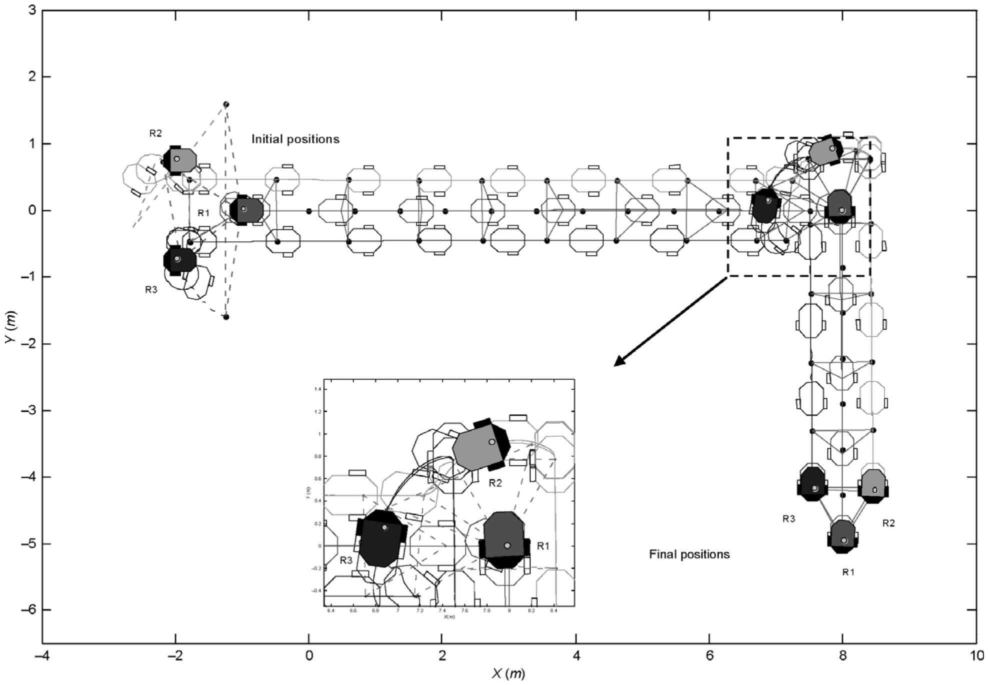

Fig. 15. Experimental trajectories of the multi-robot system with reference at right angle.

pure-rotational evolution, and the followers try to keep the formation with a significant transition error. At the end, all robots of the formation have the same heading angle. For instance, if the team of robots is transporting an object, it must be able to turn freely without dropping the transported load.

It can be observed that the proposed control system is dependent on the precision and accuracy of the sensor system; however, it is independent from the sensor method used. This relies on the fact that not only intern sensors (odometry) but also extern sensors (laser) can be used, depending on the application, complexity, or the problem to be solved. It is important to underline again that all experimental figures were reconstructed from real odometry data provided by mobile robots; in spite of this, errors introduced by odometry do not significantly affect the performance of the proposed algorithm.

6. Conclusions

In this work, a new approach to control mobile-robots formations by using linear algebra theory and numerical methods has been presented. Experimental results show that the combination of these controllers produces a straightforward and effective controller for mobile-robots formations. An appealing characteristic of this controller is its simple implementation in any programming language.

Simulation and experimental results of the developed controllers on PIONEER 3DX mobile robots have been also addressed. Through the analysis of the experiments, it can be concluded that the formation errors and the trajectory error between the desired and the real trajectory of the multi-robot system are very small. All results demonstrate satisfactory performances. The controller allows the tracking of desirable formations evolving on the plane. The use of the formation states, both in the model and in the formation

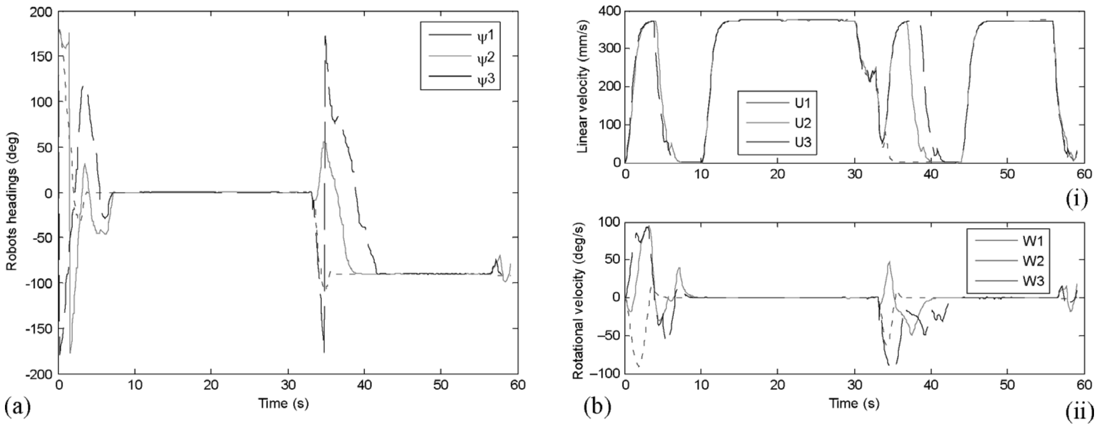

Fig. 16. (a) Headings of robots R1, R2, and R3 with reference at right angle; (b) experimental velocity profiles for R1, R2, and R3 – (i) linear; (ii) rotational.

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

control design of the multi-robot system, allows achieving a better design of the time evolution of the formation. A decentralized formation control scalable to any number of robots was presented.

In order to properly classify and develop our work, we supposed that the model of the mobile robot is a good approximation of the real system and we have considered that the uncertainties are small enough to be meaningless. Simulation and experimental results have guaranteed both hypotheses. Note that the experimental conditions were such that the errors introduced by slip and drift have a negligible impact on the difference between the actual robot position and the position estimate generated exclusively from the odometry data.

The proposed methodology for the controller design can be applied to other types of systems. The required precision of the proposed numerical method for the system approximation is smaller than the one needed to simulate the behavior of the system. This is because, when the states for the feedback are available, in each sampling time, any difference from accumulative errors is corrected (e.g., rounding errors). Thus, the approach is used to find the best way to go from one state to the next according to the availability of the system model.

Acknowledgments

This work was partially funded by the German Academic Exchange Service (DAAD – Deutscher Akademischer Austausch Dients), the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET – National Council for Scientific Research), and the Universidad Nacional de San Juan (UNSJ), Argentina. We would also like to acknowledge cooperation from the RTS – Institute for Systems Engineering, University of Hannover, Germany, especially to Prof. Dr.-Ing. Bernardo Wagner.

References

1. G. Antonelli and S. Chiaverini, "Kinematic Control of a Platoon of Autonomous Vehicles," *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (Sep. 2003) pp. 1464–1469.
2. T. Balch and R. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Trans. Rob. Autom.* **14**, 926–939 (1998).
3. T. Balch and M. Hybinette, "Social Potentials for Scalable Multi-Robot Formations," *IEEE International Conference on Robotics and Automation*, San Francisco, CA (April 2000) pp. 73–80.
4. C. Belta and V. Kumar, "Trajectory Design for Formations of Robots by Kinetic Energy Shaping," *IEEE International Conference Robotics and Automation*, Washington, DC (May 2002) pp. 2593–2598.
5. R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Rob. Autom.* **2**(1), 14–23 (1986).
6. R. Carelli, O. Nasisi, F. Roberti and S. Tosetti, "Direct Visual Tracking Control of Remote Cellular Robots," *13th. International Symposium on Measurement and Control in Robotics – Towards Advanced Robots: Design, Sensors, Control, Applications – ISMCR*, Madrid, España (Dec. 2003) pp. 11–12.
7. J. H. Chung, B. J. Yi, W. K. Kim and H. Lee, "The Dynamic Modeling and Analysis for an Omnidirectional Mobile Robot With Three Caster Wheels," *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (Sep. 2003) pp. 521–527.
8. N. Cowan, O. Shakerina, R. Vidal and S. Sastry, "Vision-Based Follow-the-Leader," *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.* **2**, 1796–1801 (2003).
9. A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer and C. Taylor, "A vision-based formation control framework," *IEEE Trans. Robot. Automat.* **18**, 813–825 (2002).
10. C. De la Cruz and R. Carelli, "Dynamic model based formation control and obstacle avoidance of multi-robot systems," *Robotica – Camb.* **26**(3), 345–356 (2008).
11. F. Del Rio, G. Jiménez, J. Sevillano, C. Amaya and A. Balcells, "Error Adaptive Tracking for Mobile Robots," *Proceedings of the 28th Annual Conference IEEE Industrial Electronics Society*, Sevilla, España (Nov. 2002) pp. 2415–2420.
12. J. P. Desai, J. Ostrowski and V. Kumar, "Controlling Formations of Multiple Mobile Robots," *IEEE International Conference on Robotics and Automation* (1998) pp. 2864–2869.
13. W. J. Dong, Y. Guo and J. A. Farrell, "Formation Control of Nonholonomic Mobile Robots," *Proceedings of American Control Conference*, Minneapolis, MN (2006), pp. 5602–5607.
14. G. Dongbing, "A differential game approach to formation control," *IEEE Trans. Cont. Syst. Technol.* **16**, 85–93 (2008).
15. R. Fierro, P. Song, A. Das and V. Kumar, "Cooperative Control of Robot Formations," **In** *Cooperative Control and Optimization*, vol. 5 (Kluwer, The Nederlands, 2002) pp. 73–93.
16. J. Fredslund and M. J. Mataric, "Robot Formations Using Only Local Sensing and Control," *IEEE International Symposium on Computational Intelligence in Robotics and Automation* (2001), pp. 308–313.
17. T. Fukao, H. Nakagawa and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Trans. Robot. Automat.* **16**, 609–615 (2000).
18. M. Hentschel, D. Lecking and B. Wagner, "Deterministic Path Planning and Navigation for an Autonomous Fork Lift Truck," *6th IFAC Symposium on Intelligent Autonomous Vehicles – IAV* (2007a).
19. M. Hentschel, O. Wulf and B. Wagner, "A hybrid feedback controller for car-like robots – combining reactive obstacle avoidance and global re-planning," *Integr. Comput.-Aided Eng.* **14**, 3–14 (2007b).
20. C. L. Hwang and L. J. Chang, "Trajectory tracking and obstacle avoidance of car-like mobile robots in an intelligent space using mixed H_2/H_∞ decentralized control," *IEEE Trans. Mechatron.* **12**(3), 345–352 (2007).
21. J. Lawton, R. Beard and B. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. Robot. Automat.* **19**, 933–941 (2003).
22. S. Lee and J. H. Park, "Virtual Trajectory in Tracking Control of Mobile Robots," *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronic*, AIM, Kobe, Japan (July 2003).
23. S. Monteiro, M. Vaz and E. Bicho, "Attractor Dynamics Generates Robot Formations: From Theory to Implementation," *Proceedings of IEEE International Conference on Robotics and Automation*, Barcelona, España (April 2004) pp. 2582–2587.
24. D. Naffin, Multi-Robot Formations: Rule-Based Synthesis and Stability Analysis *Doctoral Thesis* (University of Southern California, 2006).
25. J. Normey-Rico, I. Alcalá, J. Gomez-Ortega and E. Camacho, "Mobile robot path tracking using PID controller," *Cont. Eng. Pract.* **9**(11), 1209–1214 (2001).
26. J. Normey-Rico, J. Gomez-Ortega and E. Camacho, "A Smith-Predictor-based generalized predictive controller for mobile robot path-tracking," *Control Eng. Pract.* **7**(6), 729–740 (1999).
27. L. Ojeda and J. Borenstein, "Reduction of Odometry Errors in Over-Constrained Mobile Robots," *Proceedings of the UGV*

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press

Technology Conference at the SPIE AeroSense Symposium, Orlando, FL (April 2003) pp. 21–25.

28. W. Ren and R. W. Beard, "A Decentralized Scheme for Spacecraft Formation Flying via the Cirtual Structure Approach," *Proceedings of the American Control Conference*, Denver, CO (June 2003) pp. 1746–1751.
29. A. Rosales, G. Scaglia, V. Mut and F. di Sciascio, "Controller Designed by Means of Numeric Methods for a Benchmark Problem: RTAC (Rotational Translational Actuator)," *IEEE – Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, Mexico (Sep. 2006) pp. 97–104.
30. A. Rosales, G. Scaglia, V. Mut and F. di Sciascio, "Trajectory tracking of mobile robots in dynamic environments – A linear algebra approach," *Robot.* **27**, 981–997 (2009).
31. G. Scaglia, V. Mut, A. Rosales and O. Quintero, "Tracking Control of a Mobile Robot Using Linear Interpolation," *International Conference on Integrated Modeling & Analysis in Applied Control & Automation – IMAACA*, Buenos Aires, Argentina (Jan. 2007).
32. G. Scaglia, O. Quintero, V. Mut and F. di Sciascio, *Numerical Methods Based Controller Design for Mobile Robots* (Robotica – Cambridge University Press, Cambridge, UK, 2008a).
33. G. Scaglia, O. Quintero, V. Mut and F. di Sciascio, "Numerical Methods Based Controller Design for Mobile Robots," *IFAC World Congress*, Korea (2008b).
34. D. P. Scharf, F. Y. Hadaegh and S. R. Ploen, "A Survey of Space Formation Flying Guidance and Control (Part 2)," *Proceedings of the American Control Conference*, Boston, MA (Dec. 2004) pp. 1733–1739.
35. G. Strang, *Linear Algebra and Its Applications*, 3rd ed. (Academic Press MIT, New York, 1980).
36. K. H. Tan and M. A. Lewis, "Virtual Structures for High-Precision Cooperative Mobile Robotic Control," *Int. Conf. Intell. Robot. Syst.* **1**, 132–139 (1996).
37. T. Wang and C. Tsai, "Adaptive Trajectory Tracking Control of a Wheeled Mobile Robot via Lyapunov Techniques," *Annual Conference of IEEE Industrial Electronics Society*, Busan, Korea (Nov. 2004) pp. 389–394.
38. H. Yamaguchi, T. Arai and G. Beni, "A distributed control scheme for multiple robotic vehicles to make group formations," *Robot. Autonom. Syst.* **36**, 125–147 (2001).
39. X. Yang, K. He, M. Guo and B. Zhang, "An Intelligent Predictive Control Approach to Path Tracking Problem of Autonomous Mobile Robot," *IEEE International Conference on Robotics and Automation*, Leuven, Belgium (May 1998) pp. 3301–3306.

https://doi.org/10.1017/S0263574710000068 Published online by Cambridge University Press