5

Nonlinear Stable Formation Control using Omnidirectional Images

Christiano Couto Gava[1], Raquel Frizera Vassallo[1],
Flavio Roberti[2] and Ricardo Carelli[2]
[1]*Universidade Federal do Espítiro Santo*
[2]*Universidad Nacional de San Juan*
[1]*Brasil*
[2]*Argentina*

1. Introduction

There are a lot of applications that are better performed by a multi-robot team than a single agent. Multi-robot systems may execute tasks in a faster and more efficient way and may also be more robust to failure than a single robot. There are even some applications that can not be achieved by only one robot and just by a group of them (Parker, 2003; Cao et al., 1997). Another known advantage of multi-robot systems is that instead of using one expensive robot with high processing capacity and many sensors, sometimes one can use a team of simpler and inexpensive robots to solve the same task.

Some examples of tasks that are well performed by cooperative robots are search and rescue missions, load pushing, perimeter surveillance or cleaning, surrounding tasks, mapping and exploring. In these cases, robots may share information in order to complement their data, preventing double searching at an already visited area or alerting the others to concentrate their efforts in a specific place. Also the group may get into a desired position or arrangement to perform the task or join their forces to pull or push loads.

Although multi-robot systems provide additional facilities and functionalities, such systems bring new challenges. One of these challenges is formation control. Many times, to successfully perform a task, it is necessary to make robots get to specific positions and orientations. Within the field of robot formation control, control is typically done either in a centralized or decentralized way.

In a centralized approach a leader, which can be a robot or an external computer, monitores and controls the other robots, usually called followers. It coordinates tasks, poses and actions of the teammates. Most of the time, the leader concentrates all relevant information and decides for the whole group. The centralized approach represents a good strategy for small teams of robots, specially when the team is implemented with simple robots, only one computer and few sensors to control the entire group. In (Carelli et al., 2003) a centralized control is applied to coordinate the movement of a number of non-holonomic mobile robots to make them reach a pre-established desired formation that can be fixed or dynamic. There are also the so called leader-follower formation control as (Oliver & Labrosse, 2007; Consolini et al., 2007), in which the followers must track and follow the leader robot. The

Source: Computer Vision, Book edited by: Xiong Zhihui,
ISBN 978-953-7619-21-3, pp. 538, November 2008, I-Tech, Vienna, Austria

approach in (Oliver & Labrosse, 2007) is based on visual information and uses a set of images of the back of the leader robot that will be tracked by the follower robot. In (Consolini et al., 2007), a leader-follower formation control is introduced in which follower's position is not rigidly fixed but varies in suitable cones centered in the leader reference frame.

On the other hand, when considering a team with a large number of robots under a centralized control, the complexity significantly rises, demanding a greater computational capacity besides a larger communication system. In this case, a decentralized approach would be preferred. Usually in a decentralized control there is no supervisor and each robot makes its decisions based on its own duties and its relative position to the neighbouring teammates. Some researchers propose decentralized techniques for controlling robots' formation (Desai et al., 2001; Do, 2007) or cooperation on tasks such as exploration and mapping (Franchi et al., 2007; Correl & Martinoli, 2007; Rekleitis et al., 2005). There are also scalable approaches to control a large robotic group maintaining stability of the whole team control law (Feddema et al., 2002). Moreover some models are based on biologically-inspired cooperation and behaviour-based schemes using subsumption approach (Kube & Zhang, 1993; Balch & Arkin, 1998; Fierro et al., 2005). In these behaviour-based cases stability is often attained because they rely on stable controls at the lower level while coordination is done at a higher level.

The work presented in this chapter addresses the issue of multi-robot formation control using a centralized approach. Specifically, the principal concern is how to achieve and maintain a desired formation of a simple and inexpensive mobile robot team based only on visual information. There is a leader robot responsible for formation control, equipped with the necessary computational power and suitable sensor, while the other teammates have very limited processing capacity with a simple microcontroller and modest sensors such as wheel encoders for velocity feedback. Therefore, the team is composed of one leader and some simple, inexpensive followers. This hierarchy naturally requires a centralized control architecture. The leader runs a nonlinear formation controller designed and proved to be stable through Lyapunov theory. A nonlinear instead of linear controller was chosen because it provides a way of dealing with intrinsic nonlinearities of the physical system besides handling all configurations of the teammates, thus resulting in a more reliable option. It joins a pose controller with a compensation controller to achieve team formation and take the leader motion into account, respectively.

To control team formation it is necessary to estimate the poses of the robots that form the group. Computer vision has been used in many cooperative tasks because it allows localizing teammates, detecting obstacles as well as getting rich information from the environment. Besides that, vision systems with wide field of view also become very attractive for robot cooperation. One way of increasing the field of view is using omnidirectional images (360° horizontal view) (Nayar, 1997) obtained with catadioptric systems, which are formed by coupling a convex mirror (parabolic, hyperbolic or elliptic) and lenses (cameras) (Baker & Nayar, 1999). Such systems can really improve the perception of the environment, of other agents and objects, making task execution and cooperation easier.

Interesting works on cooperative robotics using omnidirectional images can be found in (Das et al., 2002; Vidal et al., 2004) and (Zhu et al., 2000). In (Das et al., 2002), all the robots have their own catadioptric system, allowing a decentralized strategy and eliminating the need for communication between the robots. The authors propose a framework in which a

robot can switch between controllers to follow one or two leaders, depending on the environment. However, all the processing is done on an external computer and the use of many omnidirectional systems (one for each robot) makes the team expensive. In (Vidal et al., 2004), a scenario is developed in which each follower uses optical flow for estimating the leaders relative positions, allowing the group to visually mantain a desired formation. The computational cost for optical flow calculations is high and results are shown only through simulations. The work in (Zhu et al., 2000) proposes a cooperative sensing strategy through distributed panoramic sensors on teammate robots to synthesize virtual stereo sensors for human detection and tracking. The main focus is the stereo composing and it does not address team formation.

Now, in this work, we propose a formation controller based on omnidirectional vision and nonlinear techniques that runs onboard the leader robot. To drive all followers to a specific formation, the controller considers the desired formation parameters, the leader's linear and angular velocities and current followers' poses. The desired parameters and leader velocities are assumed to be known from a higher level controller that drives the leader robot to an appropriate trajectory. The followers' poses are estimated to feedback the controller using an omnidirectional vision system, formed by a hyperbolic mirror combined with a color camera and mounted on the leader, which allows it to see all followers by acquiring just one image. It is worth mentioning that although omnidirectional vision was used to estimate followers' positions and orientations, the proposed controller is independent of which sensor is used to implement the feedback.

Followers are identified by rectangles of different colors placed on the top of their platforms. Through a set of image processing techniques such as motion segmentation and color tracking, followed by Kalman filtering combined with Least Squares and RANSAC algorithm for optimization, followers' poses are reliably estimated. These poses are then used by the nonlinear controller to define followers' linear and angular velocities to achieve and maintain the desired formation. Notice that we focus on team formation during robot motion, while obstacle avoidance and task coordination are not addressed at this stage. Simulations and real experiments were carried out with different team formations. Current results show that the system performs well even with noisy and low resolution images.

The main contribution of this work is that stable formation control is achieved based solely on visual information totally processed onboard the leader. Also, there is no need for an absolute reference frame or a limited working area, since the vision system is carried by the leader and measurements are taken relative to it. Related works usually have an expensive robot team, use a fixed camera to observe the environment or even make all computations using an external computer.

This chapter is organized as follows. Section 2 describes the formation controller. Section 3 presents a method for estimating followers' poses based on omnidirectional images. One of the simulations carried out is presented in Section 4. In Section 5, some experiments with real robots are shown and the results are discussed. Finally, Section 6 concludes this chapter and outlines the next steps.

2. The controller

To make a mobile robot team (formed by one leader and n followers) navigate in an environment keeping a specific formation, a controller to command the follower robots was designed. The leader robot coordinates group navigation using an omnidirectional system, localizing each one of the followers on its own reference frame.

2.1 Definition

A nonlinear instead of linear controller was chosen because it provides a way of dealing with intrinsic nonlinearities of the physical system besides handling all configurations of the teammates, thus resulting in a more reliable option. This controller must provide the desired values for the follower velocities based on their coordinate and orientation errors. It integrates the functions of a pose controller, that brings the team to a desired formation, and a second controller, that compensates leader's linear and angular velocities. The generated velocities are considered as reference velocities for the followers and may be sent to the robots through different means of communication. Controller stability is proved using the Lyapunov method.

This controller is similar to that found in (Roberti et al.,2007), but differs in the saturation functions for the errors, which were adapted to fit our specific problem.

2.2 The pose controller

According to Figure 1, a vector containing the followers' coordinates can be defined as Equation 1.

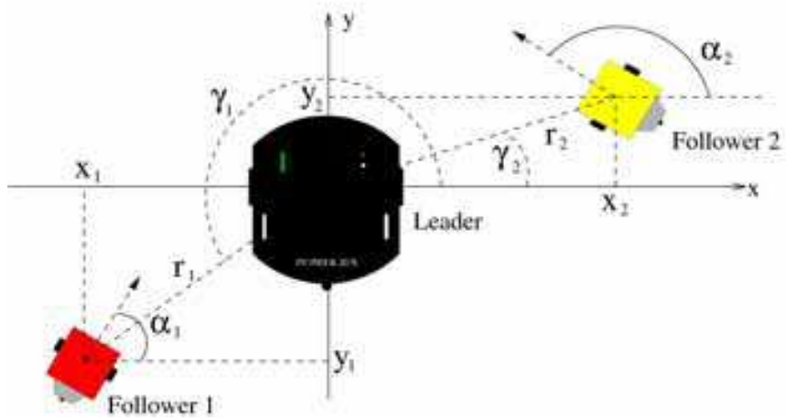


Fig. 1. Followers' pose representation on leader's reference frame.

$$\xi = \begin{pmatrix} \xi_1 & \xi_2 & \cdots & \xi_n \end{pmatrix}^T \quad \text{where} \quad \xi_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} r_i \cos(\gamma_i) \\ r_i \sin(\gamma_i) \end{pmatrix} \tag{1}$$

where $\xi_i = (x_i \quad y_i)^T$ stands for the real world coordinates of the i-th follower. To find a generic solution, the coordinate vector ξ can be considered as $\rho(\xi)$. This approach is interesting for the cases in which it is necessary to apply some coordinate transformation such as for vision systems (e.g. image coordinates) or define parameters associated to the formation (e.g. geometric parameters, baricenters, etc.). However, it is important to note that in our case $\rho(\xi) = \xi$, i. e., $\rho(\xi)$ is simply a vector containing the real world positions of the followers. We decided to keep the ρ notation for generality. By differentiating $\rho(\xi)$ with respect to time, we obtain Equation 2.

$$\dot{\rho} = J(\xi)\dot{\xi} \tag{2}$$

where $J(\xi)$ is the Jacobian of ξ.

From Equation 2, it is possible to define a formation control law given by Equation 3. The vector $\dot{\xi}_{fr}$ represents the desired formation velocities, i. e., the velocities, given at the leader reference frame, that the follower robots must have for achieving formation.

$$\dot{\xi}_{fr} = J^{-1}(\xi)\left(\dot{\rho}_d + f_{\tilde{\rho}}(\tilde{\rho})\right) \quad \text{with} \quad \tilde{\rho} = \rho_d - \rho \tag{3}$$

where $\tilde{\rho}$ is the vector of formation errors for the followers (Kelly et al., 2004) , ρ_d is the vector of desired formation parameters and ρ is the vector of the current formation parameters. Function $f_{\tilde{\rho}}(\tilde{\rho})$ is a saturation function for the error and defined as Equation 4.

$$f_{\tilde{\rho}}(\tilde{\rho}) = \text{diag}\left[\frac{k_f(\tilde{\rho}_j)}{a + |\tilde{\rho}_j|}\right]\tilde{\rho} \qquad \text{with} \qquad k_f(\tilde{\rho}_j) = k_{f1} + k_{f2}\tanh\left(|\tilde{\rho}_j|\right) \tag{4}$$

where $k_{f1} + k_{f2}$ represents the saturation value and the gain for small errors is given by k_{f1}/a. This saturation function avoids applying velocities that might saturate the robots' motors.

In Equation 3, $J^{-1}(\xi)$ is the inverse Jacobian of ξ. Computation of inverse matrices is unattractive from the point of view of efficiency. Here it becomes clear why $\rho(\xi)$ was chosen to be equal to ξ: the Jacobian of ξ is simply an identity matrix and so is the inverse Jacobian. Then $\dot{\xi}_{fr}$ is obtained through the direct sum of vectors $\dot{\rho}_d$ and $f_{\tilde{\rho}}(\tilde{\rho})$, reducing the associated computational cost.

The action of the Pose Controller is illustrated in Figure 2, where it can be seen that $\dot{\xi}_{fr}$ does not have the orientation it would if the function $f_{\tilde{\rho}}(\tilde{\rho})$ were not used. It is due to the saturation of the formation error imposed by $f_{\tilde{\rho}}(\tilde{\rho})$ which makes the sum $\dot{\rho}_d + \tilde{\rho}$, represented in Figure 2 by $\dot{\xi}'_{fr}$, different from $\dot{\rho}_d + f_{\tilde{\rho}}(\tilde{\rho}) = \dot{\xi}_{fr}$ in both norm and orientation. However, this deviation does not affect this controller's stability because as the follower approximates a desired pose, $\|\tilde{\rho}\| \to 0$ and, therefore, $\dot{\xi}_{fr} \to \dot{\rho}_d$.

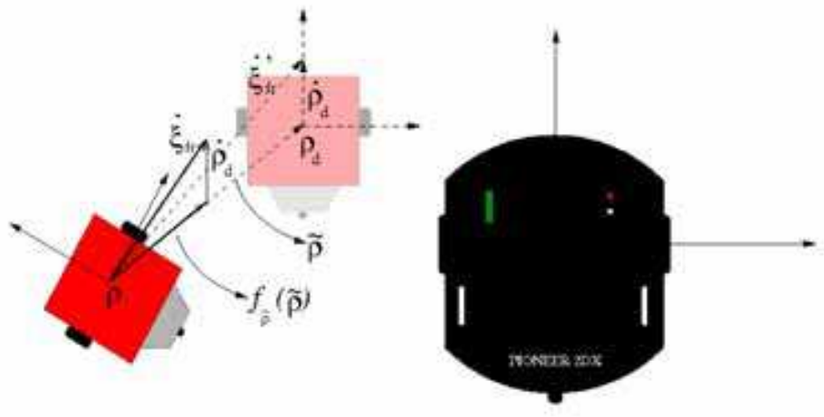


Fig. 2. Resulting $\dot{\xi}_{fr}$ after applying the Formation Controller.

Hence, the main idea for the Pose Controller is to generate velocity signals for all followers in order to bring them into formation, but taking the leader coordinate frame as reference,

which means it does not consider the leader motion. This is done by the Compensation Controller, which works in parallel with the Pose Controller.
It is known that the leader has its own linear and angular velocities, defined according to an absolute reference frame. These velocities must be considered when computing the follower velocities. In Equation 5, $\dot{\xi}_{fr}$ is added to $\dot{\xi}_{comp}$, a vector containing the compensations for the leader velocities. The resulting vector $\dot{\xi}_{r}$ provides the follower velocities needed to achieve at the same time the desired formation and compensate for the leader's motion.

$$\dot{\xi}_{r} = \dot{\xi}_{fr} + \dot{\xi}_{comp} \tag{5}$$

2.3 The compensation controller

The values of the elements of $\dot{\xi}_{comp}$ are computed to eliminate/overcome the effects caused by the leader's linear and angular velocities. Figure 3 shows an example in which the leader moves with linear (v_l) and angular (ω_l) velocities and the i-th follower is considered to be already at the desired position $(x_i \quad y_i)^T$.
Once v_l and ω_l are known, r and r_i, the circles radii described by the leader and the follower, are given by Equation 6.

$$r = \frac{v_l}{\omega_l} \text{ and } r_i = \sqrt{(r + x_i)^2 + (y_i)^2} \tag{6}$$

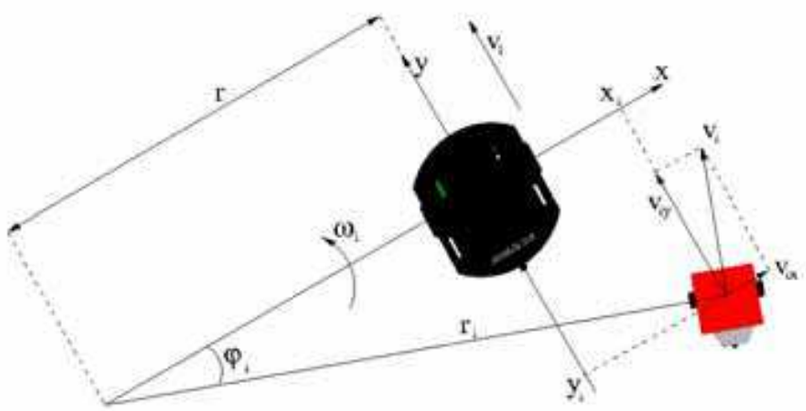


Fig. 3. Leader's velocities compensation.

Equations 7 - 9 describe the way compensation velocity is calculated for the i-th follower.

$$\varphi_i = \arctan\left(\frac{y_i}{r + x_i}\right) \text{ and } v_i = \omega_l r_i \tag{7}$$

$$v_{ix} = v_i \cos\left(\varphi_i + \frac{\pi}{2}\right) \tag{8}$$

$$v_{iy} = v_i \sin\left(\varphi_i + \frac{\pi}{2}\right) \tag{9}$$

where v_{ix} and v_{iy} are the follower compensation velocity components $\dot{\xi}_{comp}$ (v_{ix}, v_{iy}) in the leader reference frame. It is also important to mention that when the leader robot has no angular velocity ($\omega_l = 0$), $\dot{\xi}_{comp}$ equals the leader linear velocity with $v_{ix} = 0$ and $v_{iy} = v_l$.

2.4 Generating commands

After obtaining $\dot{\xi}_r$, the linear and angular velocities to be sent to the i-th robot are defined by Equations 10 and 11.

$$v_{ci} = \left\|\dot{\xi}_{ri}\right\| \cos(\tilde{\alpha}_i) \tag{10}$$

$$\omega_{ci} = \dot{\alpha}_{ri} + f_{\tilde{\alpha}}(\tilde{\alpha}_i) + \omega_l \tag{11}$$

where $\left\|\dot{\xi}_{ri}\right\|$ is the desired velocity norm for the i-th follower and $\dot{\alpha}_{ri}$ is the change in its orientation during time. The term $\tilde{\alpha}_i$, defined as $\tilde{\alpha}_i = \alpha_{ri} - \alpha_i$, is the angular error, with α_{ri} and α_i representing the reference angle and the robot current orientation, all represented in the leader frame, as shown in Figure 4. Notice that, for simplifying Figure 4 in order to help understanding, we considered the leader angular velocity (ω_l) equal to zero.
The function $f_{\tilde{\alpha}}(\tilde{\alpha}_i)$, as before, is a saturation function for the error given by Equation 12.

$$f_{\tilde{\alpha}}(\tilde{\alpha}_i) = k_{\omega 1} \tanh^3(k_{\omega 2}\, \tilde{\alpha}_i) \tag{12}$$

where $k_{\omega 1}$ represents the saturation value of the orientation error and $k_{\omega 2}$ controls how fast this function reaches its saturation value. $f_{\tilde{\alpha}}(\tilde{\alpha}_i)$ has an interesting characteristic: its derivative tends to zero as the orientation error approaches zero, which means that transitions between positive and negative values are smooth, as can be seen in Figure 5. In practice, it avoids oscilations in the followers' trajectories.

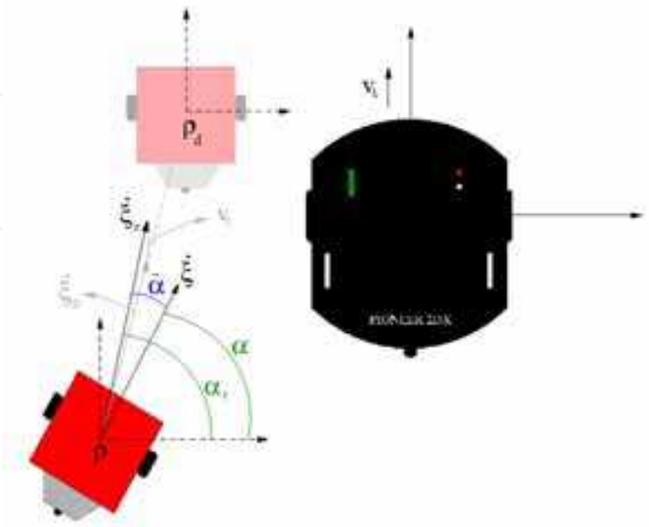


Fig. 4. Angles related to the linear and angular velocities sent to the followers.

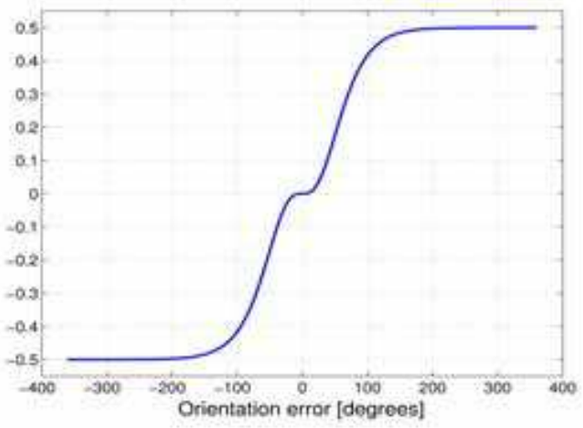


Fig. 5. Shape of $f_{\tilde{\alpha}}(\tilde{\alpha}_i)$ for $k_{\omega 1} = 0.5$ and $k_{\omega 2} = 1$.

The objective of $f_{\tilde{\alpha}}(\tilde{\alpha}_i)$ is to prevent initial orientation errors causing high angular velocity commands which would compromise control stability and submit robot motors to abrupt voltage variations.

2.5 Proof of stability

2.5.1 Proof for the pose controller

Due to their dynamics, the followers are not able to immediately achieve the desired formation velocities. However, these velocities are asymptotically achieved, represented by Equation 13, as it will be proved in Section 2.5.2.

$$\dot{\xi} \rightarrow \dot{\xi}_{fr} \tag{13}$$

where $\dot{\xi}$ is the vector of current velocities and $\dot{\xi}_{fr}$ is the vector containing the reference velocities for attaining formation. Equation 13 can also be written as Equation 14.

$$\dot{\xi}_{fr} = \dot{\xi} + \eta \quad \text{with} \quad \|\eta\| \rightarrow 0 \tag{14}$$

where η is the difference between the desired and current velocities. The control law for the Pose Controller is given by Equation 15, repeated here for convenience.

$$\dot{\xi}_{fr} = J^{-1}(\xi)\left(\dot{\rho}_d + f_{\tilde{\rho}}(\tilde{\rho})\right) \tag{15}$$

Equation 16 is obtained by the substitution of 14 in 15.

$$\dot{\xi} + \eta = J^{-1}(\xi)\left(\dot{\rho}_d + f_{\tilde{\rho}}(\tilde{\rho})\right) \tag{16}$$

Multiplying Equation 16 by $J(\xi)$ results in Equation 17.

$$J(\xi)\dot{\xi} + \eta_1 = \dot{\rho}_d + f_{\tilde{\rho}}(\tilde{\rho}) \text{ where } \eta_1 = J(\xi)\eta \tag{17}$$

As known from Equation 2, $\dot{\rho} = J(\xi)\dot{\xi}$, which leads to Equation 18.

$$\dot{\rho} + \eta_1 = \dot{\rho}_d + f_{\tilde{\rho}}(\tilde{\rho}) \tag{18}$$

The temporal derivative of $\tilde{\rho}$ produces Equation 19.

$$\dot{\tilde{\rho}} = \dot{\rho}_d - \dot{\rho} \quad \Rightarrow \quad \dot{\rho}_d = \dot{\tilde{\rho}} + \dot{\rho} \tag{19}$$

The substitution of 19 in 18 gives Equation 20.

$$\dot{\tilde{\rho}} = \eta_1 - f_{\tilde{\rho}}(\tilde{\rho}) \tag{20}$$

Then the following Lyapunov candidate function is proposed:

$$V = \frac{1}{2}\tilde{\rho}^T \tilde{\rho} \tag{21}$$

whose temporal derivative is

$$\dot{V} = \tilde{\rho}^T \dot{\tilde{\rho}} = \tilde{\rho}^T \eta_1 - \tilde{\rho}^T f_{\tilde{\rho}}(\tilde{\rho}) \tag{22}$$

For $\dot{V}$ to be definite negative it is necessary that:

$$\frac{\|k_f(\tilde{\rho})\|}{a + \|\tilde{\rho}\|}\|\tilde{\rho}\|^2 > \|\eta_1\|\|\tilde{\rho}\| \tag{23}$$

where $\|k_f(\tilde{\rho})\|$ is simply

$$\|k_f(\tilde{\rho})\| = k_f(\tilde{\rho}) = k_{f1} + k_{f2}\tanh(\|\tilde{\rho}\|) \tag{24}$$

Hence, the following condition must be satisfied:

$$\|\tilde{\rho}\| > \frac{a\|\eta_1\|}{k_{f1} + k_{f2}\tanh(\|\tilde{\rho}\|) - \|\eta_1\|} \tag{25}$$

As the followers achieve the desired velocities, $\|\eta\| \rightarrow 0$; and so $\|\eta_1\| \rightarrow 0$. Then the condition of Equation 25 will be satisfied for some finite time, which means that $\|\tilde{\rho}(t)\| \rightarrow 0$ with $t \rightarrow \infty$.

2.5.2 Proof for the generated commands

The temporal variation of the followers' orientations is expressed by

$$\dot{\alpha} = \omega_c - \omega_l \tag{26}$$

where the generated angular velocity ω_c, is given by

$$\omega_c = \dot{\alpha}_r + f_{\tilde{\alpha}}(\tilde{\alpha}) + \omega_l \tag{27}$$

Putting 27 into 26 results in Equation 28.

$$\dot{\alpha} = \dot{\alpha}_r + f_{\tilde{\alpha}}(\tilde{\alpha}) \tag{28}$$

Deriving $\tilde{\alpha}$ respect to the time gives

$$\dot{\tilde{\alpha}} = \dot{\alpha}_r - \dot{\alpha} \quad \Rightarrow \quad \dot{\alpha}_r = \dot{\tilde{\alpha}} + \dot{\alpha} \tag{29}$$

Then Equation 28 can be rewritten as

$$\dot{\tilde{\alpha}} + f_{\tilde{\alpha}}(\tilde{\alpha}) = 0 \tag{30}$$

Thus, the following Lyapunov candidate function is proposed

$$V = \frac{1}{2}\tilde{\alpha}^T \tilde{\alpha} \tag{31}$$

whose temporal derivative is

$$\dot{V} = \tilde{\alpha}^T \dot{\tilde{\alpha}} = -\tilde{\alpha}^T f_{\tilde{\alpha}}(\tilde{\alpha}) \tag{32}$$

As $f_{\tilde{\alpha}}(\tilde{\alpha})$ is an odd function, $\tilde{\alpha}^T f_{\tilde{\alpha}}(\tilde{\alpha}) > 0$ for $\tilde{\alpha} \neq 0$, which means that $\dot{V}$ is definite negative ($\dot{V} < 0$). Hence $\|\tilde{\alpha}(t)\| \to 0$ for $t \to \infty$. Finally, since $\cos(\tilde{\alpha}_i) \to 1$, we have $v_{ci} \to \|\dot{\xi}_{ri}\|$, concluding this proof.

3. Image processing and pose estimation

As said before, the leader robot is equipped with an omnidirectional vision system. Although omnidirectional images suffer from typical problems like loss of resolution and distortion, their wide field of view allows the leader to visualize all the region around itself, which facilitates localizing the teammates, avoiding obstacles and mapping the environment.

Each follower robot is identified by a colored rectangle placed on its platform. Their poses are estimated through color segmentation and Kalman filtering. Usually two colors are used on the top of the robots, so the orientation can be easily calculated (Santos-Victor et al., 2002). Because of the distortion of omnidirectional images, we decided to use just one color per robot. If two colors were used, each colored area would be reduced to half of the area seen on the image. Also image distortion increases as the robot moves away from the leader and could compromise robot localization if just a small part or none of the color of interest is seen on the image.

As discussed in the previous section, the leader must know the pose of each cellular robot belonging to the team in order for the team to navigate in formation. However, at the beginning, the leader does not know the follower's initial poses and colors. So it then needs to detect the initial position, color and orientation of each cellular robot. Once that is done the leader can start moving.

The image processing can then be divided into three main steps:

- Initial position detection;
- Tracking for initial orientation detection;
- Tracking for formation control.

In order to make the controller independent of image measurements (pixels), robot positions were converted to meters. One way of doing this and also eliminating image distortion is to

remap those images to *bird's eye view* (Vassallo et al., 2004; Pereira et al., 2005). Unfortunately this remapping depends on system calibration and requires more steps on image processing.

Instead a transform Γ, composed of a set of polynomial functions, was defined to recover robot world positions from image coordinates. The process of determining Γ is simple: first the region around the leader is divided into *n* sectors, each one defining a matching table relating distances on the image plane (pixels) and the real world (meters). Then each table is used to interpolate a polynomial function that estimates the follower positions.

Although the number of functions composing Γ can be high, it was decided to use just four, as illustrated in Figure 6, since they are enough for this application. It is important to note that this approach is much faster than using bird's eye view remapping.

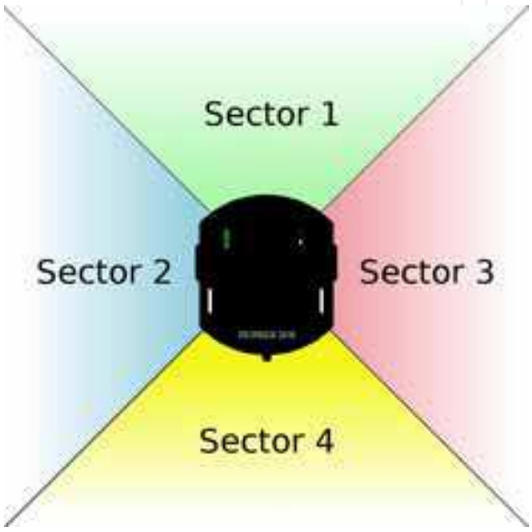


Fig. 6. Sectors used to define the Γ transform.

The polynomial functions obtained are plotted in Figure 7.

3.1 Detecting initial positions

Before starting to detect the followers' initial positions, the leader robot must focus its attention on a working area, the region around it in which all followers should be to be seen by the leader. That is because the distortion caused by omnidirectional images, which makes object detection impractical at some distance from the visual system. This region is defined by the mask exhibited in Figure 8-(a), which is applied to the omnidirectional image providing the result shown on Figure 8-(b), where a cellular robot is seen close to the leader.

This first step is accomplished by means of movement detection. Then it is not necessary to use color images, but only their grayscale version. In this work, movement detection is done based on a robust version of the background subtraction technique: instead of simply comparing a captured image with a previously constructed background, the leader compares two backgrounds. This procedure is necessary because of noise and the low resolution of omnidirectional images.

The leader starts by constructing the first background, called the *base* background -- Figure 9 -(a), while all robots are standing by. When that is finished, Follower 1 executes a short forward displacement and as soon as it stops another background is constructed, the

discriminant background -- Figure 9-(b). Then the leader compares both backgrounds and the result is thresholded, producing a blob as shown in Figure 9-(c), which represents the follower displacement and is used for estimating its initial position and color. After that the rectangle encompassing the blob -- Figure 9-(d) -- must be found because it will be used in the following steps by a tracking algorithm to estimate robot positions.

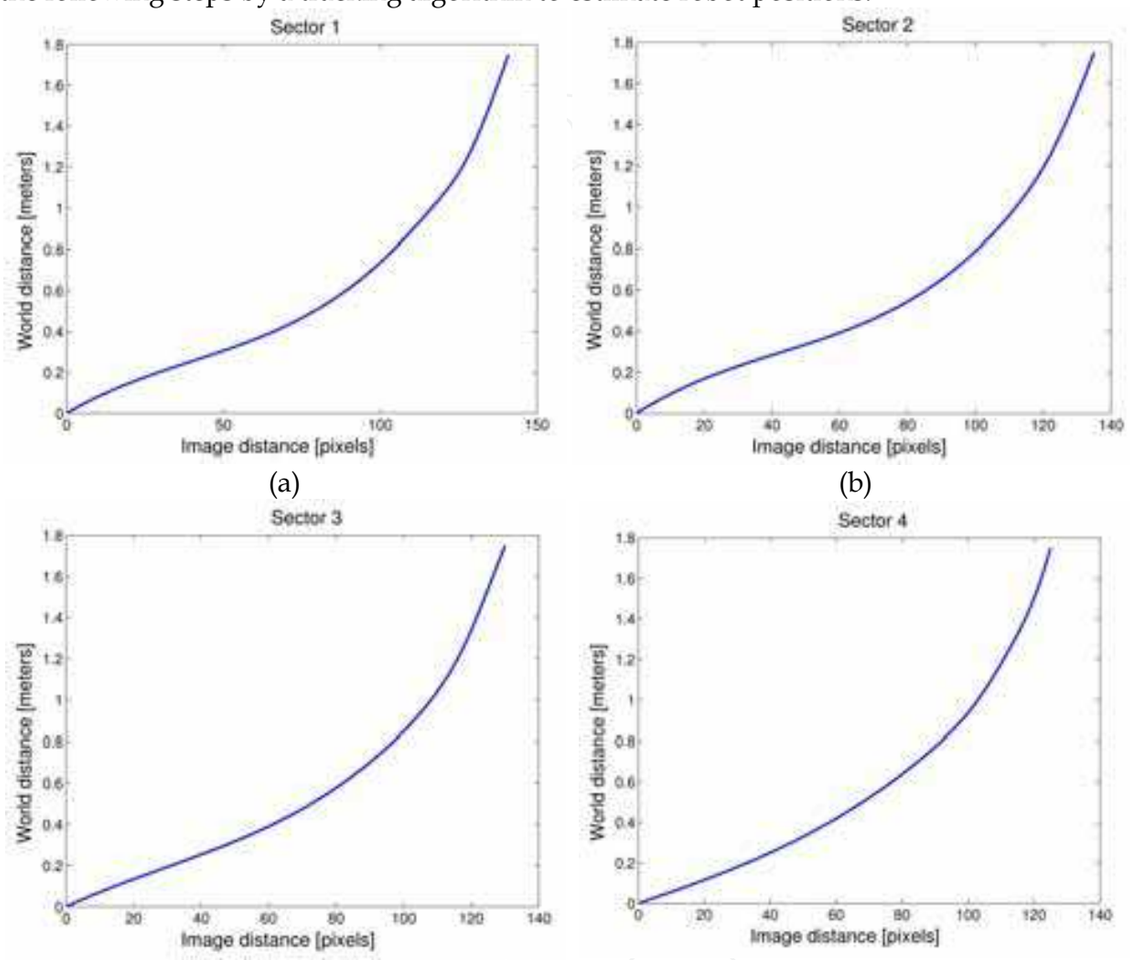
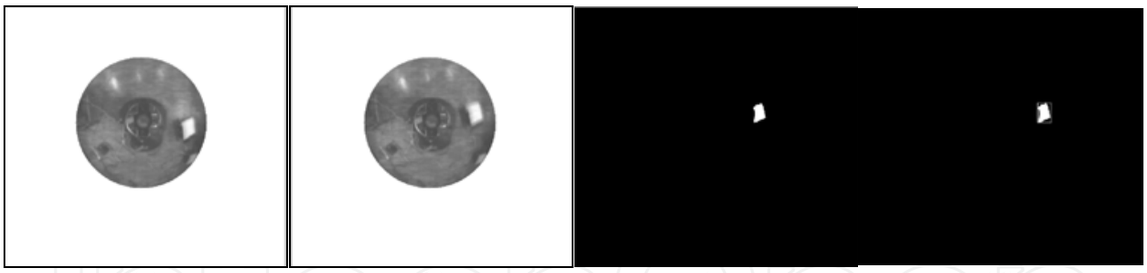


Fig. 7. Shape of each function composing the Γ transform.

Fig. 8. (a) Binary mask applied to omnidirectional image (b) Resulting working area.



(a) (b) (c) (d)

Fig. 9. (a) Base background (b) Discriminant background (c) Blob obtained after thresholding the difference between the constructed backgrounds (d) Resulting encompassing rectangle.

Often, after executing its displacement, a cellular robot generates more than one blob, as shown by Figure 10-(a). So a filter algorithm had to be developed: it checks each blob's color; if it is the same and they are sufficiently close to each other, the encompassing rectangles are combined into a single encompassing rectangle; if not, only the larger one is retained – Figure 10-(b).

(a) (b)

Fig. 10. (a) Image with more than one blob (b) Filtered image.

The above procedure is executed to detect the initial position of just one follower. However, it is not necessary to construct two new backgrounds for the next follower to be detected, since the discriminant background related to the previous robot can be used as the base background for the next one. Then, for *n* followers *n+1* backgrounds are constructed, instead of *2n*.

3.2 Detecting initial orientations

Once the color and the initial position of each follower is known, a tracking algorithm can be used for further estimate of robot positions. The CAMSHIFT (Continuously Adaptive Mean-SHIFT) algorithm, from OpenCV library, is attractive for this kind of application. Given a color histogram and an initial search window (both determined in the previous image processing step) it returns a new search window for the next image based on color segmentation. Such window is found using the dimensions of the segmented area and its centroid. This provides a fast and robust online performance for the tracking algorithm.
CAMSHIFT and Γ transform taken together allow the leader to estimate with adequate precision all follower positions. Then the procedure for finding the initial orientations is as follows: at the same time, all cellular robots execute again a forward displacement. While

they are moving, the leader saves the trajectory described by each follower, as illustrated by Figure 11. Then the leader has, for each follower, a sequence of points describing a straight line, with its angular parameter corresponding to the relative orientation α_0. However, due to measurement noises, it is necessary to apply an algorithm like RANSAC (Fischler & Bolles, 1981) to eliminate outliers. After that, a Least Squares algorithm is used to find each follower orientation thus completing the initial pose estimation.

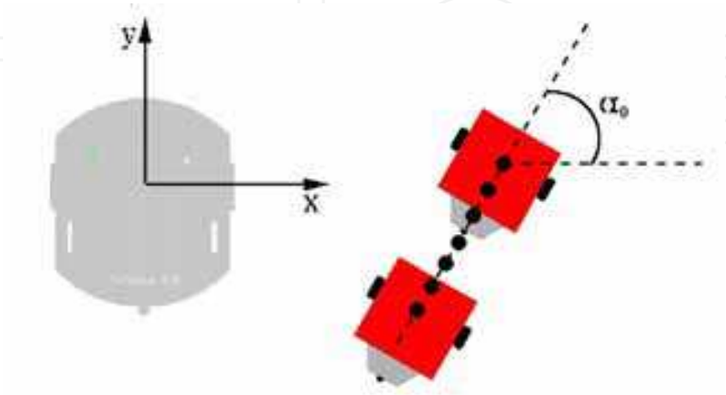


Fig. 11. Capturing a follower trajectory in order to determine its initial orientation α_0.

3.3 Tracking for formation control

Once the leader knows all followers' initial poses the logical next step would be to start moving and drive the team to the desired formation. However, there is a question that must be answered first: given a desired position, which follower should be driven to there? When a desired pose is achieved, because of the non-holonomic restriction, the respective follower must be aligned to the leader, i. e., its relative orientation must be 90°. This means that all final orientations are already known, but the final position that each follower should have is not known yet.

To solve this problem, a cost matrix C was defined - Equation 33 - where n is the number of cellular robots, c_{ij} represents the cost for the i-th follower to reach the j-th position. In other words, the i-th row of C is a vector containing the costs of the i-th follower relative to all desired positions.

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \tag{33}$$

There are many ways of defining how to calculate the costs c_{ij}. In this work, it was decided to use the square of the euclidian distance between current and desired positions. Each possible followers-desired positions configuration can be viewed as a combination of n costs, taking each one from a different row and column of C. The associated cost is simply the sum of the n costs that compose such configuration. An advantage of this approach is that it permits an analogy with the energy spent by each follower to reach some desired position. Then, it is easy to see that the ideal configuration is that having the least associated cost.

After defining the ideal configuration, the leader starts to move. To drive the cellular robots into formation it is necessary to estimate their poses, which is based on the CAMSHIFT algorithm: robot positions are estimated using the centroids of the detected colored areas and passed to the Formation Controller. However, due to measurement noises, it is very difficult to have reliable orientation values if they are estimated on every acquired image.
One way of doing that using just one color per robot is shown in (De La Cruz et al., 2006). Instead, it was decided to define a simpler method, based on the geometry of the robot trajectories, as shown in Figure 12. Each follower orientation is calculated after the robot has moved at least a minimum displacement Δs_{min}, defined in Equation 34 (these values were chosen empirically), whose objective is to reduce noise influence caused by image low resolution, mirror distortion and illumination changes.

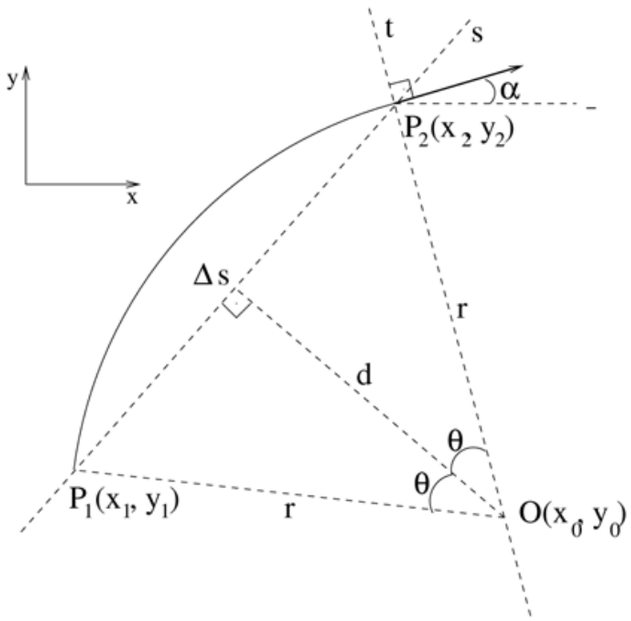


Fig. 12. Follower trajectory while its pose and velocities are not updated.

$$\Delta s_{min} = 0.02 + 0.03\tanh\left(2\|\tilde{\rho}\|\right) \tag{34}$$

The orientation α is estimated considering that, between two control signal updates, the robot maintains the previous linear and angular velocities and performs a curve trajectory. From Figure 12, the straight line s can be defined by Equation 35.

$$\begin{aligned} &y = m_s x + l_s \quad \text{with} \\ &m_s = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{and} \quad l_s = y_2 - m_s x_2 \end{aligned} \tag{35}$$

The distance d is obtained from the displacement Δs and the angle θ, using the follower's angular velocity ω and the time interval Δt spent to move from $P_1(x_1, y_1)$ to $P_2(x_2, y_2)$.

$$d = \frac{\Delta s}{2\tan(\theta)} \quad \text{where} \quad \theta = \frac{\omega \Delta t}{2} \tag{36}$$

Then d, the line s and the circle equation are used to find $O(x_0, y_0)$, which is used to calculate the angle α.

$$\alpha = \arctan\left(\frac{-1}{m_t}\right) \quad \text{where} \quad m_t = \frac{y_2 - y_O}{x_2 - x_O} \tag{37}$$

A special case is considered when $x_2 = x_1$: if $y_1 > y_2$, $\alpha = -\pi/2$, if not, $\alpha = \pi/2$. Then a Kalman Filter was applied to α, resulting in more stable estimates and reducing the errors for the next loop control. Kalman filtering was chosen because of its performance and low computational cost.
Figure 13 shows follower motion detection for the first pose estimation and the further tracking software running. White outlines involve the colorful rectangles segmented from an omnidirectional image.

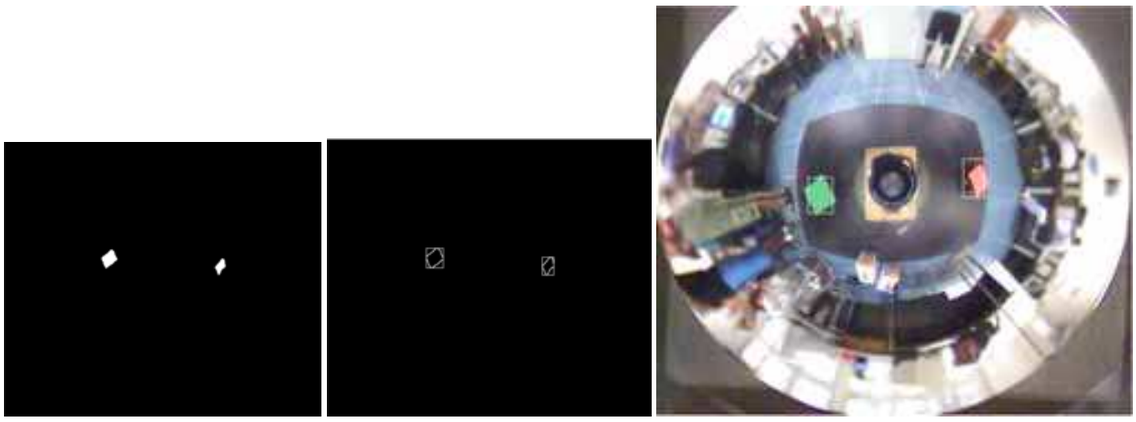


Fig. 13. Robots detection and the tracking software.

From the above equations it is clear that Δs must be known in order to determine robot orientations. It is computed while the team is moving and this is done based on a 2D geometric transformation, which is defined by composing a translation and a rotation, since the leader has, in general, both linear and angular velocities. The idea is to define two reference frames: the first corresponds to where the previous image was captured – S_0 – and the second to where the current image has been captured – S_1 – according to Figure 14, with dx and dy standing for x and y leader displacements and γ its rotation.
Then, knowing x_0, y_0, dx, dy and γ, Equation 38 shows how to obtain x_1 and y_1. It is important to note that $(x_1 \quad y_1)^T$ do not mean the current follower position, but the previous position represented in the most recent frame S_1. Δs can now be calculated through the euclidian distance between the current position and $(x_1 \quad y_1)^T$.

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -dx \\ 0 & 1 & -dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{38}$$

Due to the projection geometry of omnidirectional visual systems, robot orientations are not affected by the translation of the coordinate system, only by its rotation. Figure 15 shows this effect, where α_0 and α_1 stand for a robot orientation on previous and current reference

frame, respectively. Hence, it is easy to see that $\alpha_1 = \alpha_0 - \gamma$. Every time a follower presents a Δs greater than Δs_{min} its pose should be updated and passed to the controller in order to generate new control signals.

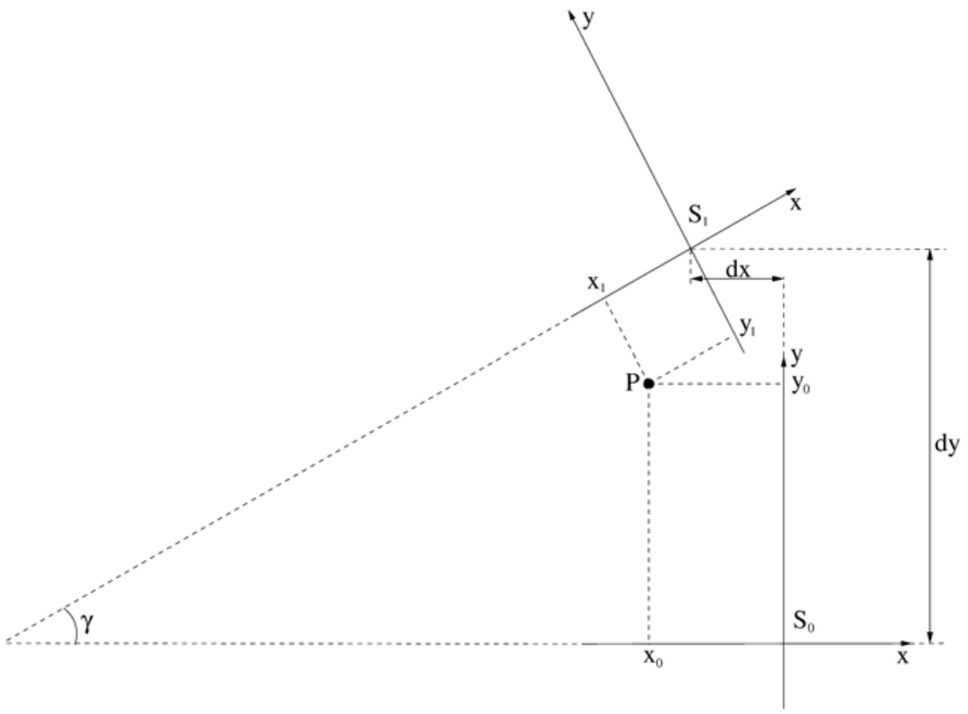


Fig. 14. Effects of leader's frame translation and rotation on a follower position representation.

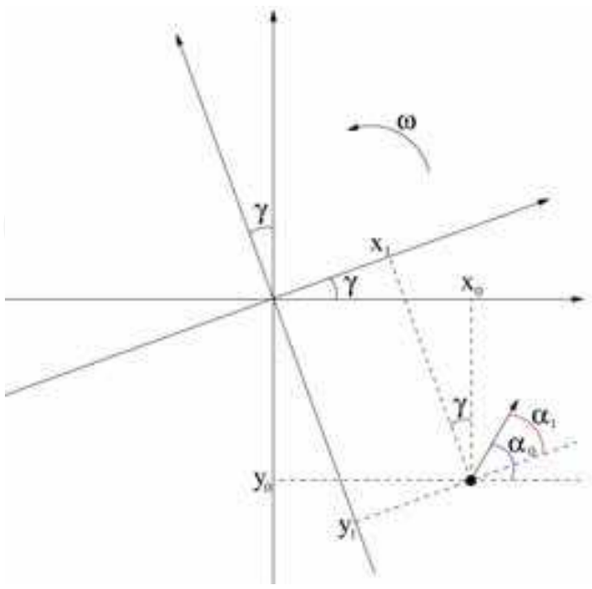


Fig. 15. Effect of leader's frame rotation over a follower orientation.

4. Simulation

Before testing on real robots, some simulations were carried out to evaluate the behavior of the proposed controller while varying some parameters, although not considering the team dynamics. Without loss of generality, the initial position of the leader is chosen to be coincident with the world frame origin. Several simulations were carried out with the same controller parameters used in the experiments.

The idea of the presented simulation is to reproduce an obstacle avoidance maneuver while maintaining formation, a common situation in navigation tasks. Figure 16 shows the trajectories described by the team during this simulation. The leader had a linear velocity of 60 mm/s and an angular velocity according to the function in Equation 39, which is the same function used in the last experiment shown in this chapter.

$$f_{\omega l}(t) = \begin{cases} k_1 \tanh(k_2 (t_1 - t)) & \text{for} \quad 10 \le t \le 100s \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

where k_1, k_2 and t_1 are auxiliary parameters used to control the desired shape of the leader's trajectory. For this simulation $k_1 = 1.5°/s$, $k_2 = 0.2$ and $t_1 = 55$ s, which means that for $10 < t < 100$ s the leader's angular velocity varied from -1.5°/s to 1.5°/s, reaching zero at $t = t_1 = 55$ s. The middle blue line indicates the leader's trajectory. The dashed outer lines represent the desired trajectories for the followers, that must stay on positions $\rho_{d1} = (-0.50 \quad -0.30)^T$ and $\rho_{d2} = (0.50 \quad -0.30)^T$ relative to the leader. The solid lines indicate the followers' trajectories that started from initial positions $\rho_{01} = (-0.70 \quad -0.80)^T$ and $\rho_{02} = (0.30 \quad -0.90)^T$. The red triangles indicate how the team gets into formation. Followers' initial orientations were 135° and 120°, respectively. The followers achieve their desired positions and successfully maintain the formation, describing the proposed trajectory.

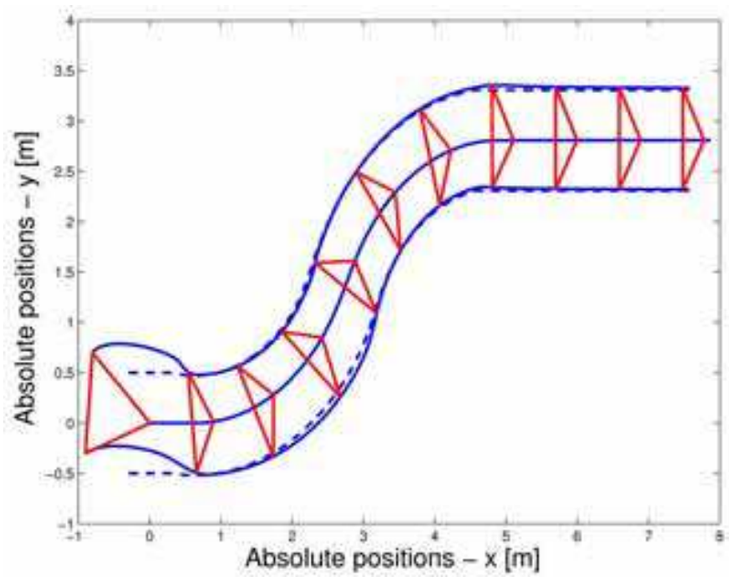


Fig. 16. Trajectory described by the team during this simulation.

Figure 17 exhibits the effect of the leader's angular velocity variation on followers' poses, in which (a) and (b) indicate followers' position errors while (c) shows their orientations. It is

possible to see that abrupt variations or sign changes of the leader's angular velocity disturb the system, thus causing perturbations on followers' poses just after they happen. Since such perturbations are common during robot navigation, the controller must drive all followers to the desired poses regardless of that, which is shown by Figures 16 and 17.

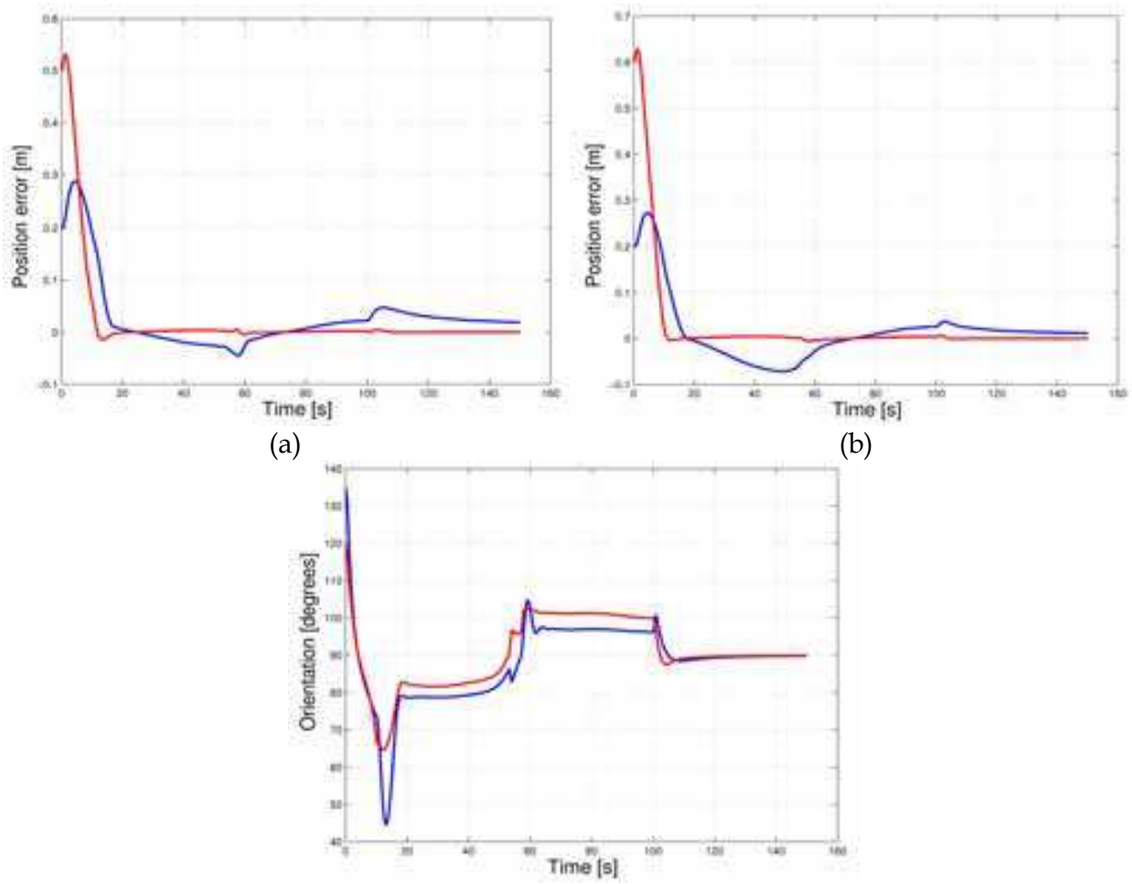


Fig. 17. Simulation results: position errors for (a) Follower 1 (b) Follower 2. (c) Followers orientations.

5. Experiments and results

The experiments presented here were performed with a robot team composed of a Pioneer 2DX (Pentium II, 266 MHz, 128 MB RAM) as leader and two cellular robots as followers. They are shown in Figure 18.

The leader has an omnidirectional system with a perspective color camera and a hyperbolic mirror. The two cellular robots were assembled in our lab and have about the size of 15 x 25 cm and 10 cm height. They are differential robots equipped with the MSP430F149 microcontroller and H-bridges TPIC0108B from *Texas Instruments* to drive the motors. Initially communication between leader and followers was accomplished by cables for serial communication, substituted later by a radio link.

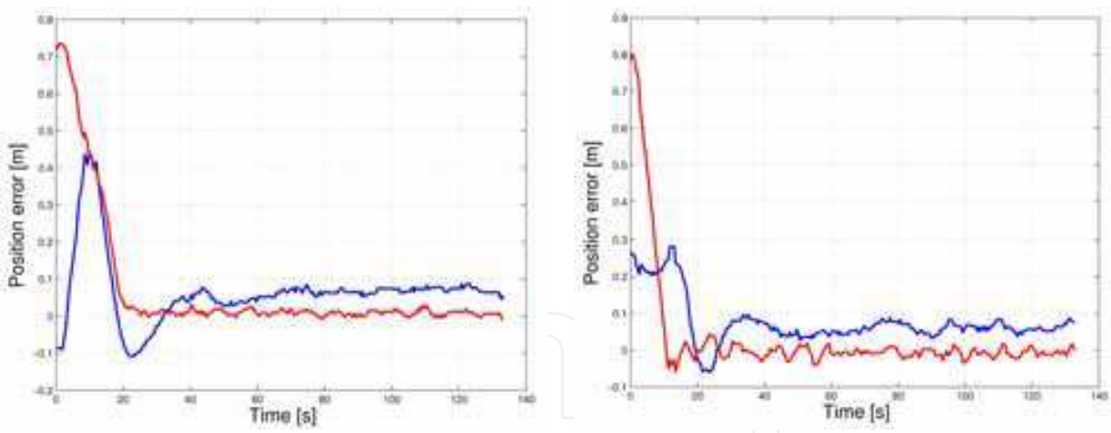
Several experiments were also carried out. We decided to present three of them because they show the key features of the proposed controller.

Fig. 18. Robots used for the experiments.

5.1 Experiment 1

In this first experiment, the idea was to execute a simple translation in order to show the controller behaviour in a free corridor, for example. Then the leader has developed a straight line with 60 mm/s of velocity. Followers' initial positions were estimated at ρ_{01} = (-0.50 -0.40)^T, ρ_{02} = (0.35 -0.50)^T, while the desired positions were ρ_{d1} = (-0.60 0.30)^T, ρ_{d2} = (0.60 0.30)^T, that is, an isoceles triangle with followers in front of the leader. Initial orientations were approximately 135° and 60°, respectively.

Figure 19 shows the position errors for both followers, while Figure 20 gives the behaviour of their orientations. The error related to the x-coordinate is plotted in blue, while the red line indicates the error for the y-coordinate. In Figure 20, the calculated values for orientation are indicated in blue and the resultant values after Kalman Filtering are in red.

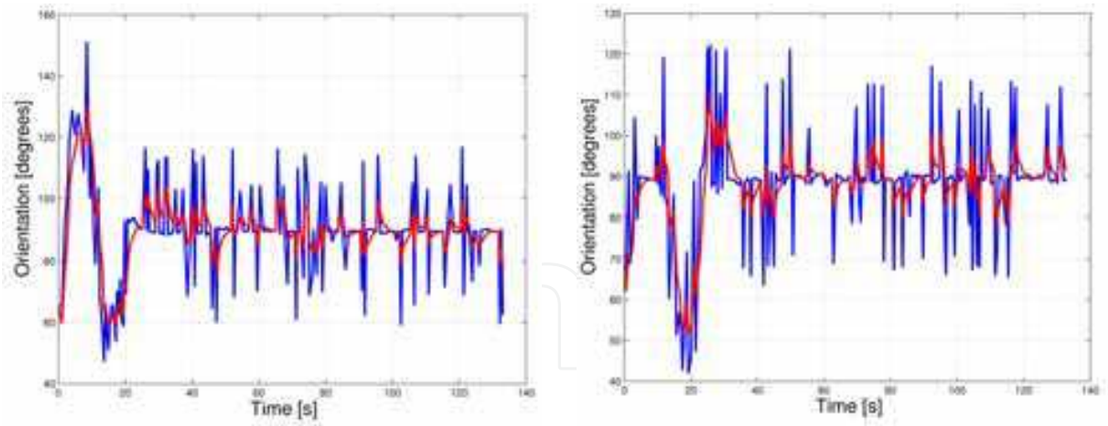


(a) (b)

Fig. 19. Experiment 1: position errors (blue) x and (red) y for (a) Follower 1 (b) Follower 2.

From Figure 20 it is possible to see that after the transient stage, both orientations stay close to 90°. This result, together with the position errors shown in Figure 19, means that both followers achieve the desired poses and keep them until the end of the experiment.

The trajectory described by the group is illustrated in Figure 21, in which it is clear that it is not actually straight. The reason is a disalignment on the leader's left wheel, which affects encoder readings. As a result, the robot is unable to perform exactly the required linear and angular velocities. This effect is present in all experiments shown, but becomes more visible when the trajectory should be a straight line.



(a) (b)

Fig. 20. Experiment 1: orientation behaviour (blue) before and (red) after filtering of (a) Follower 1 (b) Follower 2.

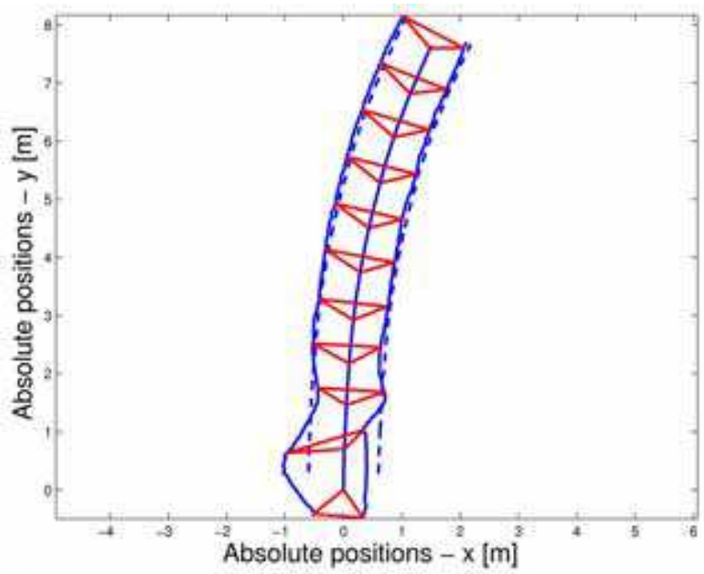


Fig. 21. Experiment 1: trajectory performed by the group.

5.2 Experiment 2

As has been shown in the first experiment, the leader navigated with fixed orientation. Now, the idea is to evaluate the controller behaviour when the reference frame is always changing its orientation. This second experiment was run with this purpose. Once again, leader's linear velocity was 60 mm/s, but its angular velocity was constant and equal to 1,5 °/s.

The followers were detected at positions given by $\rho_{01} = (-0.70 \quad 0.0)^T$, $\rho_{02} = (0.0 \quad -0.90)^T$ and their desired positions were $\rho_{d1} = (0.0 \quad 0.70)^T$, $\rho_{d2} = (0.0 \quad -0.70)^T$, which means that one follower should stay in front of the leader and the other behind it, as shown by Figure 22. Initial orientations were both estimated as being 85°.

Figure 22 serves also to see that the radii described by the followers are greater then that described by the leader and they are related by $r_i = \sqrt{r^2 + y_i^2}$, with i = 1, 2. The position

errors obtained are depicted in Figure 23 and Figure 24 shows the evolution of both followers' orientations.

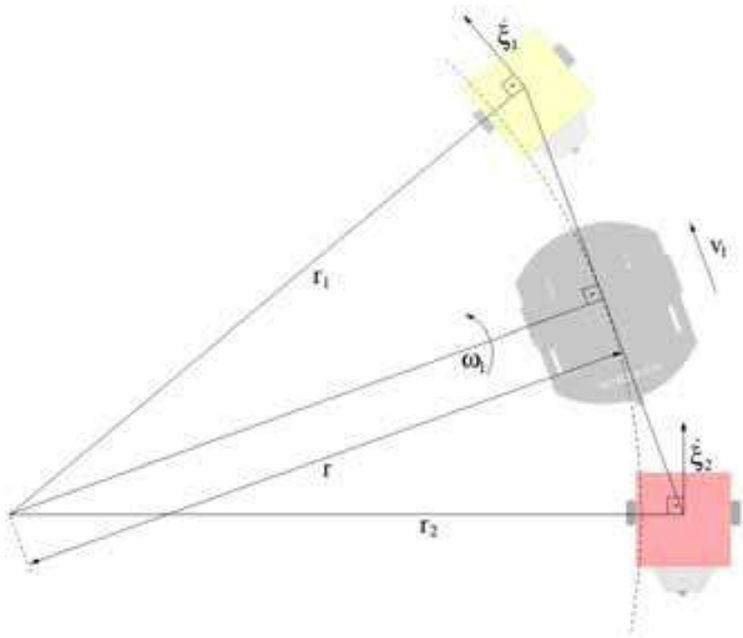
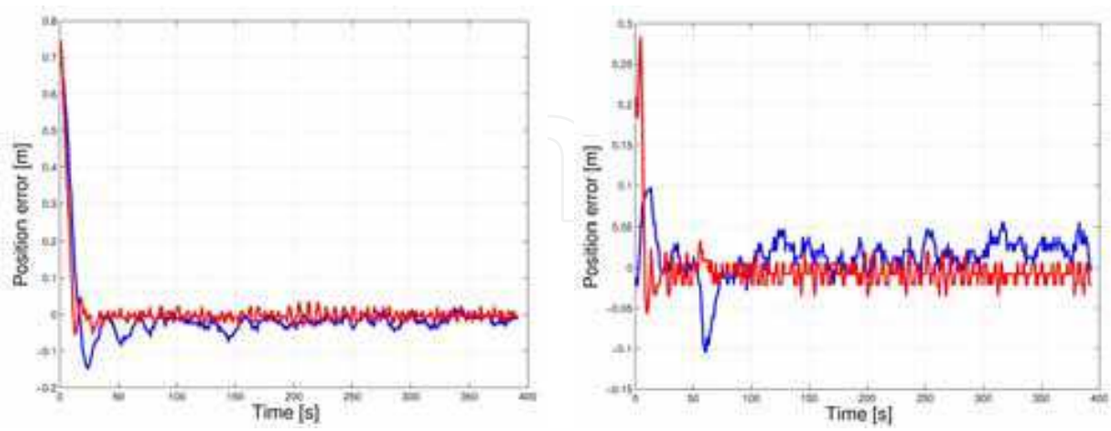


Fig. 22. Expected geometry formation for the second experiment.

Here it is worth mentioning the reason for the negative peak relative to the second follower's x-error (Figure 23-(b)) after about 60 seconds of experiment: this robot got stuck for a moment and could not follow the leader. But as soon as it could move again, the controller brought it back to the desired pose. We decided to present this particular experiment also because it shows the robustness of the controller on dealing with disturbances.

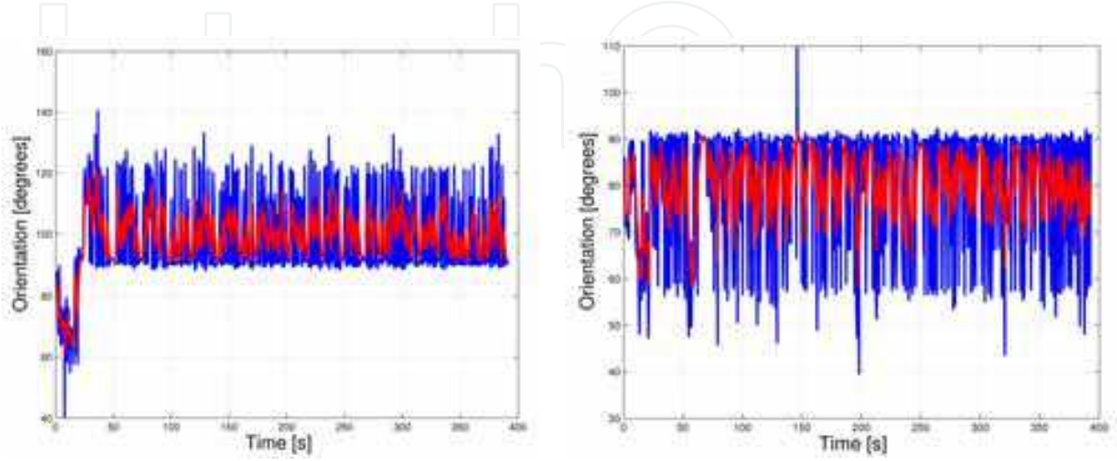


(a) (b)

Fig. 23. Experiment 2: position errors (blue) x and (red) y for (a) Follower 1 (b) Follower 2.

Another important observation can be done with respect to the orientations presented by the cellular robots. According to Figure 24 the followers' orientations did not achieve the

steady state around 90°, but close to 100° and 80°, respectively. This fact was already expected and can be easily explained by Figure 22, where it can be seen that for a counterclockwise rotation the robot that is going in front of the leader must have an angle greater than 90° relative to the leader frame, while the other robot must present an orientation less than 90°. As might be expected, the relation between these angles is inverted for a clockwise rotation.



(a) (b)

Fig. 24. Experiment 2: orientation behaviour before (blue) and after filtering (red) of (a) Follower 1 (b) Follower 2.

The resulting trajectory is illustrated in Figure 25 in which the team was moving counterclockwise, since leader's angular velocity was positive. It should be noted that the robots rapidly achieved the desired formation and maintained it until closing the circle, as shown by the triangle representing team formation, almost becoming a straight line.

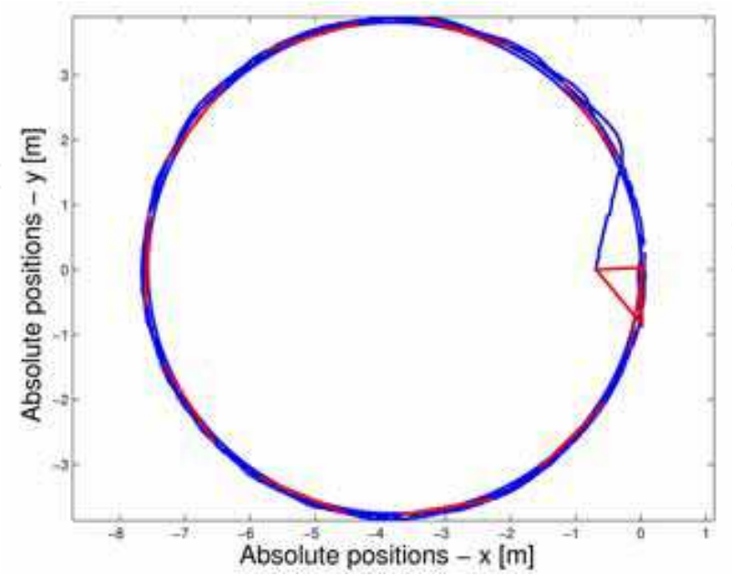
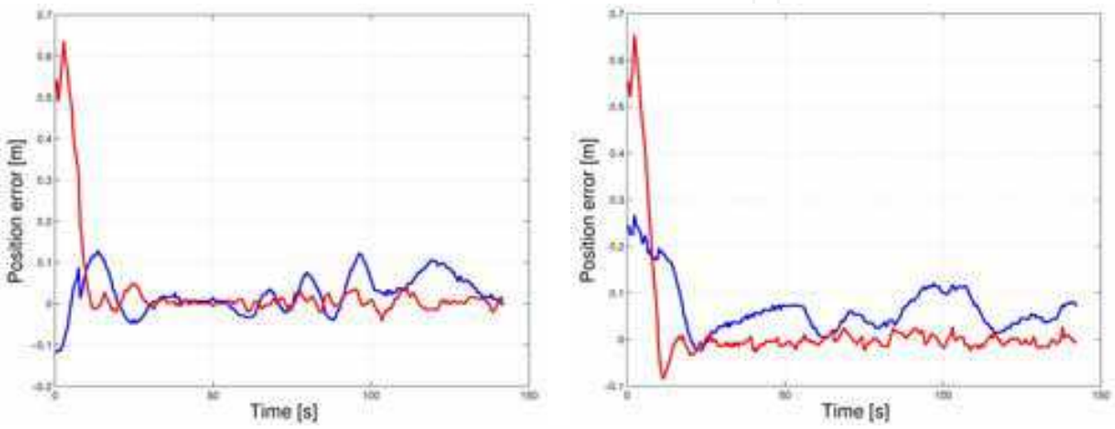


Fig. 25. Experiment 2: trajectory performed by the group.

5.3 Experiment 3

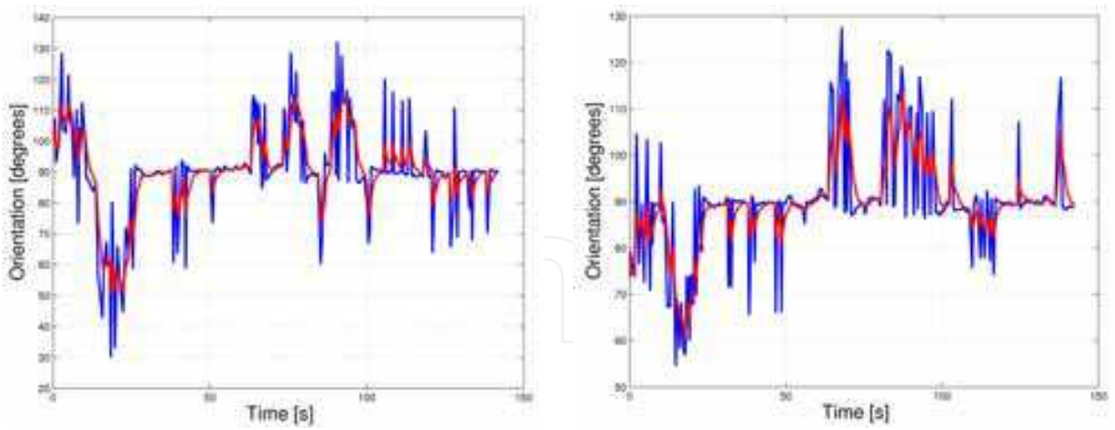
The simulation presented above, although not considering robots' dynamics, illustrates the controller behaviour in a common situation in which the team needs to get into formation and avoid an obstacle at the same time. The objective of this last experiment was to evaluate the controller in the same situation, but using real robots.

Hence, in this experiment the leader navigated with the same velocities it had in the simulation and followers' initial positions were approximately $\rho_{01} = (-0.40 \quad -0.80)^T$ and $\rho_{02} = (0.25 \quad -0.85)^T$, while their desired positions were $\rho_{d1} = (-0.50 \quad -0.30)^T$ and $\rho_{d2} = (0.50 \quad -0.30)^T$. Initial orientations were estimated as being 105° and 80°, respectively. Figure 26 gives the position errors and Figure 27 shows the followers' orientations obtained with this experiment.



(a) (b)

Fig. 26. Experiment 3: position errors (blue) x and (red) y for (a) Follower 1 (b) Follower 2.



(a) (b)

Fig. 27. Experiment 3: orientation behaviour before (blue) and after filtering (red) of (a) Follower 1 (b) Follower 2.

As in the simulation, followers' poses suffered from leader's angular velocity variations, but the controller successfully drove the robots to the desired formation. Figure 28 shows the performed trajectory, which is not exactly the same of that obtained in the simulation because of the disalignment on the leader's left wheel.

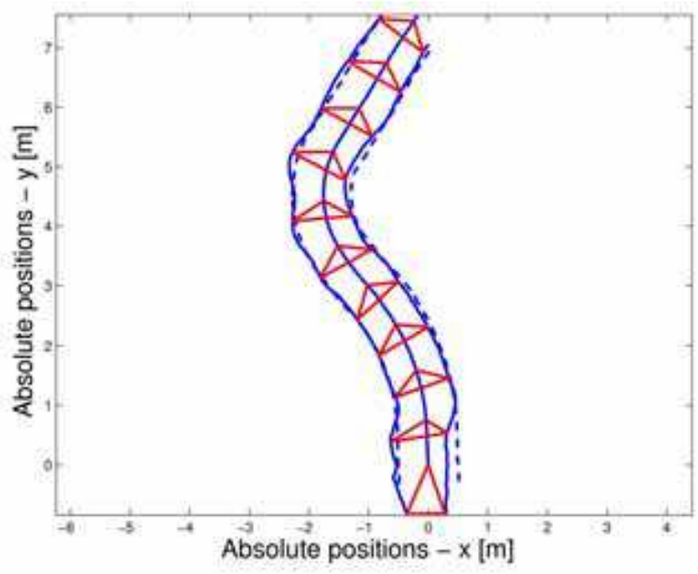


Fig. 28. Experiment 3: trajectory performed by the group.

5.4 Remarks

The experiments shown demonstrate that the robot team has achieved the desired formation and maintained it until the end of the respective experiment, even suffering the influence of image noise, low resolution of the camera and reduced useful image area. Position errors were limited to 10 cm in most experiments.

The use of Kalman Filter provided more reasonable orientation values, thus significantly improving the controller performance through better robot's pose estimation. As a result, the generated commands are smoother than those obtained without filtering.

6. Conclusion and future work

This chapter has presented a multirobot formation control strategy based on nonlinear theory and omnidirectional vision. The objective is to drive a team of simple and inexpensive mobile robots to a desired formation using only visual information. Because of the limitations of the celular robots they must be led by a leader robot having the necessary processing capacity and equipped with an adequate sensor. Thus, the formation control is done using a centralized approach.

Group formation is accomplished by a stable nonlinear controller designed to drive the followers into formation during navigation regardless of which sensor is used to implement the feedback. In this work, feedback was implemented using a single omnidirectional vision system because it allows the leader to localize all followers by acquiring just one image.

An important advantage of our approach is that the working area is not limited since the vision system is attached to the leader and so is the reference frame, which means all measurements are taken relative to the leader. Besides that, all computations are carried out onboard the leader, discarding the use of an external computer.

Through a set of image processing techniques followers' poses are reliably estimated. That includes motion segmentation, morphological filtering and color tracking, complemented by

Kalman filtering combined with Least Squares and RANSAC algorithm for optimization. Followers' positions and orientations are then used by the controller to define desired velocities for the robots to attain formation.
Simulations and experiments were carried out to evaluate the controller performance and current results show that the system performs well even with noisy and low resolution images. As future work, the controller shall be improved and obstacle avoidance will be included. Optical flow on omnidirectional images might play an important role on obstacle avoidance, and time to collision can be used to provide a safe team navigation.
Finally, this work may represent a good step towards applications that require using a large number of robots while keeping costs within reason. Combining centralized and decentralized strategies could be used to make a large robot group divide itself into smaller teams each having one leader. Leaders would negociate navigation and task execution among themselves while controlling the follower teammates. This approach would provide stable formation control and robustness against the failure of a leader. In this case, other leaders could adopt the "orphan" followers and the task in charge would be handled with reduced impact.

7. Acknowledgment

The authors would like to thank FAPES, through the projects FUNCITEC 30897360/2005 and FUNCITEC 38425025/2007, CAPES (Brazil) and SPU (Argentina), through the binational program CAPG-BA, for their financial support. A partnership between UFES, Brazil, and INAUT-UNSJ, Argentina, allowed Christiano Couto Gava to stay three months in San Juan, and Flavio Roberti to stay three months in Vitória, where part of this work was developed. It is also important to thank Fabricio Bortolini de Sá and Marino Frank Cypriano (undergraduate students) for their commitment on assembling the follower robots and Celso De La Cruz Casaño for his assistence.

8. References

Parker, L. E. (2003). Current Research in Multirobot Systems. Artificial Life Robotics, Vol. 7, No. 1-2 (March 2003) pp. 1-5, ISSN 1433-5298.

Cao, Y. U.; Fukunaga, A. S.; Kahng, A. B. (1997). Cooperative Mobile Robotics: Antecedents and Directions. Autonomous Robots, Vol. 4, No. 1 (March 1997), pp. 7-27, ISSN 0929-5593.

Carelli, R.; De La Cruz, C.; Roberti, F.(2006). Centralized formation control of non-holonomic mobile robots. Latin American Applied Research, Vol.36, No.2 (2006), pp.63-69.

Oliver, J.; Labrosse, F. (2007). Towards an appearance-based approach to the leader-follower formation problem, Proceedings of Taros 2007 - Towards Autonomous Robotic Systems, pp. 137-144, United Kingdom, September, 2007.

Consolini, L.; Morbidi, F.; Prattichizzo, D.; Tosques, M. (2007). A Geometric Characterization of Leader-Follower Formation Control, Proceedings of ICRA 2007 - IEEE International Conference on Robotics and Automation, pp. 2397-2402, ISBN 1-4244-0601-3, Roma, Italy, April 2007.

Desai, J. P.; Ostrowski, J. P.; Kumar, V. (2001). Modeling and Control of Formations of Nonholonomic Mobile Robots, IEEE Transactions on Robotics and Automation, Vol. 17, No. 6 (December 2001), pp. 905-908, ISSN 1042-296X.

Do, K. D. (2007). Bounded Controllers for Decentralized Formation Control of Mobile Robots with Limited Sensing, International Journal of Computers, Communications & Control, Vol. 2, No. 4 (2007), pp 340-354.

Franchi, A.; Freda, L.; Oriolo, G.; Venditteli, M. (2007). A Randomized Strategy for Cooperative Robot Exploration, Proceedings of ICRA 2007 - IEEE International Conference on Robotics and Automation, pp. 768-774, ISBN 1-4244-0601-3, Roma, Italy, April 2007.

Correl, N.; Martinoli, A. (2007) Robust Distributed Coverage using a Swarm of Miniature Robots, Proceedings of ICRA 2007 - IEEE International Conference on Robotics and Automation, pp. 379-384, ISBN 1-4244-0601-3, Roma, Italy, April 2007.

Rekleitis, I.; Peng New, A.; Choset, H. (2005) Distributed Coverage of Unknown/Unstructured Environments by Mobile Sensor Networks, In: Multi-Robot Systems. From Swarms to Intelligent Automata. Volume III, Editors Parker, L.; Schneider, F. E.; Schultz, A. C., pp. 145-155, Springer, ISBN 978-1-4020-1185-6.

Feddema, J. T.; Lewis, C.; Schoenwald, D. A. (2002). Decentralized Control of Cooperative Robotic Vehicles: Theory and Application, IEEE Transactions on Robotics and Automation, Vol. 18, No. 5 (October 2002), pp. 852-864, ISSN 1042-296X.

Kube,R. C.; Zhang, H. (1993). Collective robotics: From social insects to robots, Adaptive Behavior, Vol. 2, No. 2 (1993), pp. 189-218, ISSN1059-7123.

Balch, T.; Arkin, R. C. (1998). Behavior-based formation control for multi-robot teams, IEEE Transactions on Robotics and Automation, Vol. 14, No. 6 (December 1998), pp. 926-939, ISSN: 1042-296X.

Fierro, R.; Clark, J.; Hougen, D.; Commuri, S. (2005). A Multi-Robot Testbed for Biologically-Inspired Cooperative Control, In: Multi-Robot Systems. From Swarms to Intelligent Automata. Volume III, Editors Parker, L.; Schneider, F. E.; Schultz, A. C., pp. 171-182, Springer, ISBN 978-1-4020-1185-6.

Nayar, S. K. (1997). Omnidirectional Vision, Proceedings of the 8th International Symposium on Robotics Research, Hayama, Japan, October 1997.

Baker, S.; Nayar, S. K. (1999). A Theory of Single-Viewpoint Catadioptric Image Formation, International Journal of Computer Vision, Vol. 35, No. 2 (November 1999), pp. 175-196, ISSN 0920-5691.

Das, A. K.; Fierro, R.; Kumar, V.; Ostrowski, J. P., Spletzer, J.; Taylor, C. J. (2002). A Vision-Based Formation Control Framework, IEEE Transactions on Robotics and Automation, Vol. 18, No. 5 (October 2002), pp. 813-825, ISSN: 1042-296X.

Vidal, R.; Shakernia, O.; Sastry, S. (2004). Following the Flock, IEEE Robotics & Automation Magazine, Vol. 11, No.4 (December 2004), pp. 14-20, ISSN 1070-9932.

Zhu, Z.; D. Rajasekar, K.; Riseman, E. M.; Hanson, A. R. (2000). Panoramic Virtual Stereo Vision of Cooperative Mobile Robots for Localizing 3D Moving Objects, Proceedings of IEEE Workshop on Omnidirectional Vision - OMNIVIS'00, pp. 29-36, ISBN 0-7695-0704-2, Hyatt Regency, USA, June 2000.

Roberti, F.; Toibero, J. M.; Carelli, R.; Vassallo, R. F. (2007). Stable formation control for a team of wheeled mobile robots, Proceedings of XII Reunión de Trabajo en Procesamiento de la Información y Control, 6 p., Rio Gallegos, Argentina, 2007.

Kelly, R.; Carelli, R.; Zannatha, J. M. I.; Monroy, C. (2004). Control de una Pandilla de Robots Móviles para el Seguimiento de una Constelación de Puntos Objetivo, Proceedings of VI Congreso Mexicano de Robótica, pp. 83-89, COMRob, Torreón, Coahuila, México, 2004.

Santos-Victor, J. A.; Carelli, R.; Van Zwaan, S. (2002). Nonlinear Visual Control of Remote Cellular Robots, Proceedings of the10th Mediterranean Conference on Control and Automation, Lisbon, Portugal, July 2002.

Vassallo, R. F.; Encarnação, L. F.; Santos-Victor, J. A.; Schneebeli, H. J. A. (2004). Bird's Eye View Remapping and Path Following Based on Omnidirectional Vision, Proceedings of XV CBA - Congresso Brasileiro de Automática, Gramado, Brasil, September 2004.

Pereira, F. G.; Gava, C. C.; Vassallo, R. F.; Sarcinelli-Filho, M. (2005). Calibração de Sistemas Catadióptricos e Detecção da Pose de Robôs Móveis por Segmentação de Imagens Omnidirecionais, Proceedings of VII SBAI - Simpósio Brasileiro de Automação Inteligente, 8p., São Luís, Brasil, September 2005.

Fischler, M. A.; Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM, Vol. 24, No.6 (June 1981), pp. 381-395.

De La Cruz, C.; Carelli, R.; Gava, C. C. (2006). Control Centralizado de Formación Usando Una Cámara Omnidireccional, Proceedings of IV Jornadas Argentinas de Robótica, 6p., Córdoba, Argentina, November 2006.

Computer Vision
Edited by Xiong Zhihui

ISBN 978-953-7619-21-3
Hard cover, 538 pages
Publisher InTech
Published online 01, November, 2008
Published in print edition November, 2008

This book presents research trends on computer vision, especially on application of robotics, and on advanced approachs for computer vision (such as omnidirectional vision). Among them, research on RFID technology integrating stereo vision to localize an indoor mobile robot is included in this book. Besides, this book includes many research on omnidirectional vision, and the combination of omnidirectional vision with robotics. This book features representative work on the computer vision, and it puts more focus on robotics vision and omnidirectioal vision. The intended audience is anyone who wishes to become familiar with the latest research work on computer vision, especially its applications on robots. The contents of this book allow the reader to know more technical aspects and applications of computer vision. Researchers and instructors will benefit from this book.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen