2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Nonlinear Control Techniques and Omnidirectional Vision for Team Formation on Cooperative Robotics

Christiano Couto Gava, Raquel Frizera Vassallo, Flavio Roberti, Ricardo Carelli and Teodiano Freire Bastos-Filho

Abstract— **In this work a robot cooperation strategy based on omnidirectional vision is presented. Such strategy will be applied to a mobile robot team formed by small and simple robots and a bigger leader robot with more computational power. The leader must control team formation. It has an omnidirectional camera and sees the other robots. Color segmentation and Kalman filtering is used to obtain the pose of the followers. This information is then used by a nonlinear stable controller to manage team formation. Simulations and some preliminary experiments were run. The current results are interesting and encourage towards the next steps.**

I. Introduction

Since the first mobile robots applications started to be presented, people expect that robots play tasks such as house keeping, elderly assistence, environment mapping, monitoring, inspecting, exploring, rescuing and many others. Usually these tasks demand robots with high processing capacity and many sensors in order to deal and interact with the environment and other agents, which normally represents expensive costs. However there are some of those applications that may be performed by a team of simpler and cheaper robots, in a faster and more efficient way than a single robot.

Some examples of tasks that are better performed by cooperative robotics are search and rescue missions, load pushing, perimeter surveillance, surrounding tasks, mapping and exploring [1], [2], [3]. While moving, the robots may share information in order to complement their data, preventing double searching at an already visited area or otherwise alerting the others to concentrate their efforts in a specific place, or even guiding the group to a desired positon or formation.

Many times, to successfully perform a task, it is necessary to control team formation so the robots can get to right positons and orientations. Control of cooperative robotic can be done in a centralized or decentralized way. In both cases, to get into formation and to keep it there is the assignment of one or more leader robots so each robot in the team follows its corresponding leaders.

There are researches that propose decentralized control techniques for formation of nonholomic robots [4], and also a scalable approach to large group of robots keeping the stability of the whole team control law [2]. There are also some models based on animal-like cooperation behavior and behavior-based schemes using subsumption approach [5], [6]. Usually in these behavior-based cases stability is reached because they rely on stable controls at the lower level while coordination is done at a higher level.

To control team formation it is necessary to have a good estimation of the robots pose. Computer vision has been used in many cooperative tasks because it allows localizing teammates, detecting obstacles and also getting rich information from the environment. Besides that, vision systems with wide field of view also become very attractive for robot cooperation. Such systems can really improve the perception of the environment, of other agents and objects, making task execution and cooperation easier.

One way of increasing the field of view is using omnidirectional images (360^o of horizontal view) [7] obtained with catadioptric systems, which are formed by coupling a convex mirror (parabolic, hyperbolic or elliptic) and lenses (cameras) [8].

Interesting works on cooperative robotics using omnidirectional images can be found in [9], [10] and [11]. In [9], all the robots have their own catadioptric system, allowing a decentralized strategy and eliminating the need of communication between the robots. It proposes a framework where 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 [10], a scenario is developed where each follower uses optical flow for estimating the leaders relative positions, allowing the group to visually mantain a desired formation. The computacional cost for optical flow calculations is high and results are shown only through simulations. The work in [11] proposes a cooperative sensing strategy through distributed panoramic sensors on teammate robots to synthesize virtual stereo sensors for human detection and tracking. The work focus the stereo composing and do not address the team formation control.

Now, in this paper, we propose a formation strategy based on omnidirectional vision and nonlinear control techniques. The approach uses just one omnidirectional camera and control stability is proved by the Lyapunov method. Once the team is composed by very simple robots, a centralized formation control is used by the robot leader, which has greater computational power and own the omnidirectional system. Although the formation control is centralized on the leader, the followers velocities control is done in a

This work was supported by FAPES and CAPES (Brazil) and SPU (Argentina)

decentralized manner. Once the followers receive their motor commands sent by the leader, each one has its own controller for performing the asked linear and angular velocities.

Basically the leader is in charge of team navigation and task coordination. It has a catadioptric system formed by a hyperbolic mirror and a color camera. The followers have a simpler microcontroller and modest sensors such as wheel encoders for velocity feedback. These robots are identified by different colorful rectangles placed on the top of their plataform. From omnidirectional images, the leader estimates the followers poses through color segmentation and Kalman filtering. Those poses are then used for controlling team formation through a stable nonlinear controller that defines the followers linear and angular velocities to achieve and keep the desired formation.

In this paper we focus on the team formation during robots motion. Obstacle avoidance and task coordination will not be addressed at this stage of our work. This paper is organized as follows. Section 2 presents robot pose estimation based on omnidirectional images. Section 3 describes the formation controller. In Section 4, some experiments are shown and the results are discussed. Finally, Section 5 presents the conclusions and future work.

II. Image Processing and Pose Estimation

Omnidirectional images allow the leader robot to visualize all the region around itself, which facilitates localizing the other team robots. Each follower robot is identified by a colorful rectangle placed on its plataform. Their poses are estimated through color segmentation and Kalman filtering.

Usually two colors are used on the robots, so the orientation can be easily calculated [12]. Because of the distortion on omnidirecional images, we decided to keep with just one color per robot. If two colors were used, each colorful 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. That could spoil robot localization if just a small part or none of the color of interest is seen on the image.

To start tracking the followers, the leader orders the teammates to move. Moving areas are segmented through background subtraction and morphology filtering is performed for reducing noise influence. Then edge detection is done and the resulting contours are used for finding the colors of each region. The images, given on the RGB space, are converted to the HSV space and the hue channel is then used for extracting the color of each detected region. Each different color corresponds to one follower and is used for further color segmentation by a tracking algorithm, in order to track the robots motion and estimate their poses on the next images. The tracking software was implemented based on CAMShift algorithm from the OpenCV library.

Robots positions are estimated using the centroids of the detected colorful areas and passed to the formation controller. In order to make the controller independent of image measurements (on pixels), robots 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* [13], [14]. Unfortunately this remapping depends on system calibration and represents more steps on image processing.

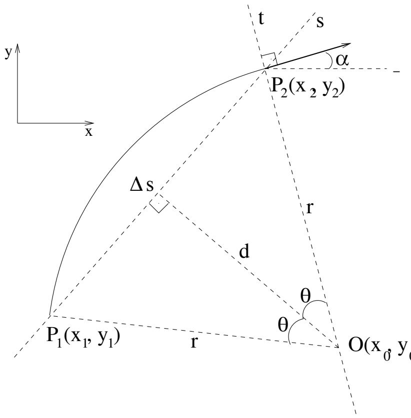


Fig. 1. Follower trajectory while its pose and velocities are not updated.

Instead, a polynomial function that recovers robots world positions from image centroids was defined. A matching table, relating distances on the image plane (pixels) and the real world (meters), was used to interpolate a polynomial function that estimates the followers positions with less than 5 cm of error. This approach is much faster than using bird's eye view remapping.

Once robots positions are estimated, is time to calculate the orientations. One way of doing that using just one color per robot is shown in [15]. Instead, we decided to define a simpler method, based on the robots trajectories geometry, as shown in Figure 1. Each follower orientation is calculated after the robot moves at least 5 cm[1]. That reduces noise influence caused by image low resolution, mirror distortion and illumination changes. The orientation α is calculated considering that, between two control signal updates, the robot keeps the previous linear and angular velocities and performs a curve trajectory.

From Figure 1, the straigh line s can be defined by Equation 1.

$$\begin{aligned} y &= m_s x + l_s \quad \text{with} \\ m_s &= \frac{y_2 - y_1}{x_2 - x_1} \quad \text{and} \quad l_s = y_2 - m_s x_2 \end{aligned} \tag{1}$$

The distance d is obtained from the displacement Δs and the angle θ, using the robot angular velocity ω and the time interval Δt while moving from $P_1\,(x_1, y_1)$ to $P_2\,(x_2, y_2)$.

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

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

$$\alpha = \arctan\left(\frac{-1}{m_t}\right) \quad \text{where} \quad m_t = \frac{y_2 - y_0}{x_2 - x_0} \tag{3}$$

[1]This value was chosen after some trials with different displacement values.

A special case is considered when $x_2 = x_1$: if $y_1 > y_2$, $\alpha = \frac{-\pi}{2}$, if not, $\alpha = \frac{\pi}{2}$. In order to reduce noise influence, we applied a Kalman Filter to make better estimations on every control loop. That improves the calculated orientation values and helps reducing the errors for the next estimations.

Figure 2 shows the followers motion detection for the first pose estimation and the further tracking software running. White outlines involve the colorful rectangles segmented from an omnidirectional image. The calculated centroid and area size are used for generating a search window for the next incoming image. That provides a fast and robust online performance for the tracking algorithm.

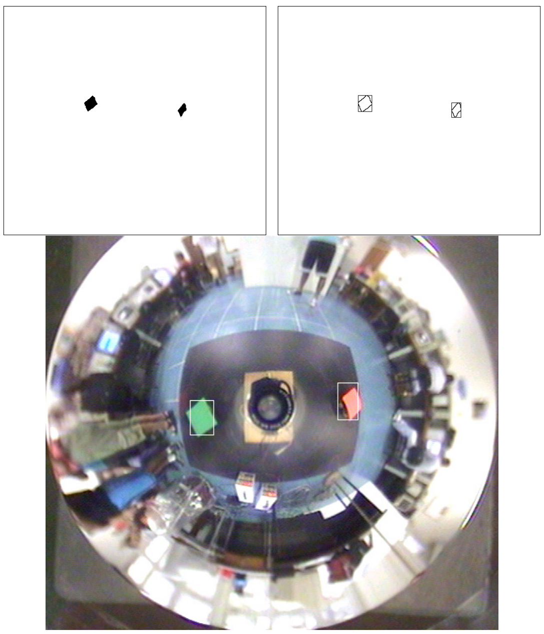


Fig. 2. Robots detection and the tracking software.

The estimated poses are already represented on the leader reference frame because estimation was done using omnidirectional images. Then robots poses are used by a nonlinear controller to keep team formation while the leader moves. With the obtained precision, the commands sent by the controller keeps the system stability.

III. The Controller

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

A. Definition

A nonlinear controller for the followers was defined based on their pose information and the intrinsic nonlinearity that describes the system. The intention is to have a stable controller that works at any operation point and not just around a specific point (as in the case of linearized systems). This controller must provide the desired values for the followers velocities based on their coordinate and orientation errors. It joins a formation controller, that brings the team to a desired formation, and an additional controller, that compensates the leader linear and angular velocities [16]. The generated velocities are considered as reference velocities for the followers and may be sent to the robots through different ways of communication [12]. Controller stability is proved using the Lyapunov method.

B. Theoretical description

A vector containing the followers coordinates can be defined as Equation 4.

$$\xi = \begin{pmatrix} \xi_1 & \xi_2 & \dots & \xi_n \end{pmatrix}^T \tag{4}$$

where $\xi_i = (x_i \;\; y_i)^T$ stands for the real world coordinates of the i-th follower. To find a generic solution, the coordinates vector ξ can be considered as $\rho(\xi)$. This approach is interesting for the cases where 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.). Then, by differentiating $\rho(\xi)$ by time, we obtain Equation 5.

$$\dot{\rho} = J(\xi)\,\dot{\xi} \tag{5}$$

where $J(\xi)$ is the Jacobian of ξ.

From Equation 5, it is possible to define a formation control law given by Equation 6. 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.

$$\dot{\xi}_{fr} = J^{-1}(\xi)\left(\dot{\rho}_d + f_{\tilde{\rho}}(\tilde{\rho})\right) \;\text{ with }\; \tilde{\rho} = \rho_d - \rho \tag{6}$$

where $\tilde{\rho}$ is the vector of formation errors for the followers [17], 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 on the error and defined as Equation 7.

$$f_{\tilde{\rho}}(\tilde{\rho}) = diag\left[\frac{k_f}{a + |\tilde{\rho}_j|}\right]\tilde{\rho} \tag{7}$$

where k_f represents the saturation value and a is such that $\frac{k_f}{a}$ represents the gain for small errors. This saturation function avoids applying velocities that might saturate the robots actuators.

However, the leader has its own linear and angular velocities, defined according to an absolute reference frame. These velocities must be considered when computing the followers velocities. In Equation 8, $\dot{\xi}_{fr}$ is added to $\dot{\xi}_l$ and $\dot{\xi}_\omega$, which are leader velocities compensations. The resulting vector $\dot{\xi}_r$ provides the followers velocities needed to achieve at the same time the desired formation and compensate the leader's motion.

$$\dot{\xi}_r = \dot{\xi}_{fr} + \dot{\xi}_l + \dot{\xi}_\omega \tag{8}$$

The values for $\dot{\xi}_l$ and $\dot{\xi}_\omega$ are calculated to compensate the effects caused by the leader linear and angular velocities. Figure 3 shows an example where the leader moves with linear (v) and angular (ω) velocities and the i-th follower is considered to be already at the desired position $(x_i \quad y_i)^T$.

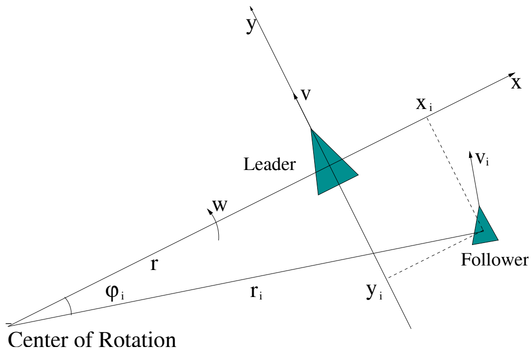


Fig. 3. Leader's velocities compensation.

Once v and ω are known, r and r_i, the circles radii described by the leader and the follower, are given by Equation 9.

$$r = \frac{v}{\omega} \quad \text{and} \quad r_i = \sqrt{(r+x_i)^2 + (y_i)^2} \tag{9}$$

Equations 10 - 12 describe the way compensation velocity is calculated for the i-th follower. These equations represent the contributions of both $\dot{\xi}_l$ and $\dot{\xi}_\omega$ in Equation 8.

$$\varphi_i = arctan\left(\frac{y_i}{r+x_i}\right) \quad \text{and} \quad |v_i| = \omega r_i \tag{10}$$

$$|v_{ix}| = |v_i|\cos\left(\varphi_i + \frac{\pi}{2}\right) \tag{11}$$

$$|v_{iy}| = |v_i|\sin\left(\varphi_i + \frac{\pi}{2}\right) \tag{12}$$

where v_{ix} and v_{iy} are the follower compensation velocity components in the leader reference frame.

After obtaining the compensations needed for getting into formation and following the leader, the linear and angular velocities to be sent to the i-th robot are defined by Equations 13 and 14.

$$\dot{\xi}_{ci} = \|\dot{\xi}_{ri}\|\cos(\tilde{\alpha}_i) \tag{13}$$

$$\omega_{ci} = \dot{\alpha}_{ri} + f_{\tilde{\alpha}}(\tilde{\alpha}_i) + \omega \tag{14}$$

where $\|\dot{\xi}_{ri}\|$ is the desired velocity norm for the i-th follower and $\dot{\alpha}_{ri}$ is the change on its orientation during time. The term $\tilde{\alpha}_i = \alpha_{ri} - \alpha_i$ is the angular error, α_{ri} is the reference angle and α_i is robot current orientation, all represented in the leader frame. The function $f_{\tilde{\alpha}}(\tilde{\alpha}_i)$, just as before, is a saturation function on the error given by Equation 15.

$$f_{\tilde{\alpha}}(\tilde{\alpha}_i) = k_\omega tanh(\tilde{\alpha}_i) \tag{15}$$

where k_ω represents the saturation value on the orientation error. The function $f_{\tilde{\alpha}}(\tilde{\alpha}_i)$ aims to prevent that initial orientation errors cause high angular velocities commands. That may compromise control stability and submit robot motors to abrupt voltage variations.

C. Simulations

Before testing on real robots, some simulations were carried out to evaluate the proposed controller. Simulating allows to observe the controller behavior while varying some parameters, although it does not consider the team dynamics. Without loosing generality, the leader initial position is chosen coincident with the world frame origin. Figure 4 shows a simulation where the leader has a linear velocity of 10 mm/s and angular velocity according to the function $\omega(t) = 0.05tanh(t-b)$, where b is an auxiliary parameter used to control the desired trajectory's shape.

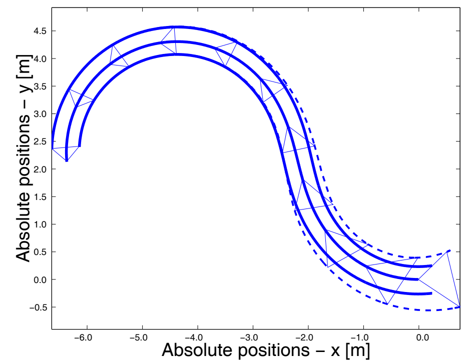


Fig. 4. Simulation of the nonlinear controller.

The middle line indicates the leader trajectory. The solid outer lines represent the desired trajectories for the followers, that must stay on positions $x_1 = -0.25$, $y_1 = -0.25$ and $x_2 = 0.25$, $y_2 = -0.25$ related to the leader. The dashed lines indicate the followers trajectories that started from initial positions $x_{01} = 0.75$, $y_{01} = -0.50$ and $x_{02} = 0.50$, $y_{02} = 0.50$. The drawn triangles indicate how the team gets into formation. The leader and the followers initial orientation were π, $\frac{\pi}{6}$ and $\frac{-3\pi}{4}$ rad. Simulations were done with the same controller parameters used in the experiments. The followers achieve their desired positions and successfully keep the formation, describing the proposed trajectory.

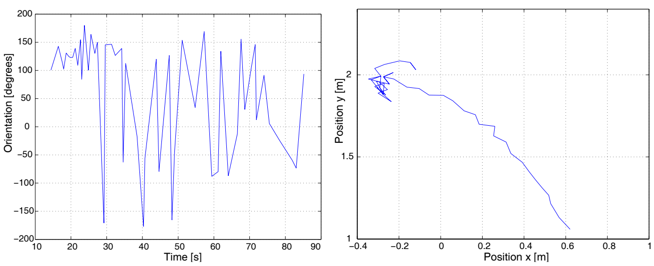
IV. Experiments and Results

The experiments presented in this paper were performed with a robot team composed by a Pioneer 2DX robot (Pentium II, 266 MHz, 128 MB RAM) as the leader and two celular robots as the followers (see Figure 5). The leader has an omnidirectional system composed by a perspective color camera and a hyperbolic mirror. The two celular robots were assembled in our lab and have about the size of 15 x 25 cm and 10 cm of height. They are differential robots using the MSP430F1611 microcontroller and H-bridges TPIC0108B from *Texas Instruments* for driving the motors. Because we do not have the radio link ready yet on the two celular robots, leader-follower communication was accomplished by a cable link. We already bought *ZigBee* modules and now we are looking forward to have the radio link installed for implementing the communication between the robots.

First, some experiments were done to test the method used for estimating the followers orientation. Figures 6 and 7 show

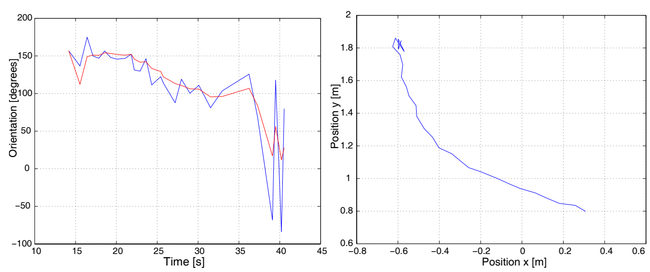
Fig. 5. Robots used for the experiments.

two trials where the formation controller guided a follower to get into a desired pose. The experiment in Figure 6 does not include Kalman filtering while Figure 7 does. Filtering improves significantly the pose estimation and helps the formation control, contributing to get better motor commands that result on a more soft trajectory for the follower.



(a) (b)

Fig. 6. Without filtering: (a) Orientation estimatives. (b) Follower trajectory.



(a) (b)

Fig. 7. With filtering:(a) Orientation estimatives (on blue: no filter, on red: with filter). (b) Follower trajectory.

On Figure 7-(a), the abrupt changes at the last estimations are due bad encoders readings at low velocities, when the robot is almost arriving at the desired pose and stoping.

After that some initial experiments on group formation were carried out. One of them was chosen to be presented in this paper. The experiment last 2 minutes and 20 seconds from the initial steps of followers segmentation and pose estimation until getting and keeping group formation. The results are shown in the following Figures 9 - 12.

A. The Experiment

The leader moved with linear velocity of 60 mm/s. The desired positions for the followers on the leader's frame correspond to the coordinates $x_{d1} = -0.7$ and $y_{d1} = 0$, $x_{d2} = 0.7$ and $y_{d2} = 0$, which put the follower robot-1 on the left side of the leader and robot-2 at the right side, as shown in Figure 8.

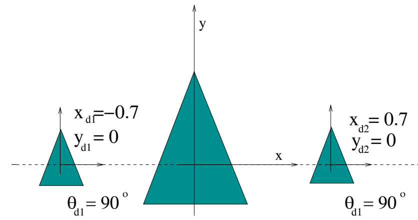


Fig. 8. Desired formation.

Figure 9 shows the robot-1 position errors while Figure 10 does the same for robot-2. The solid line represents the x error coordinate while the dashed line shows the y error. For both robots, the position errors approach zero. For robot-1 they are less than 5 cm and for robot-2, the x error is less than 10 cm. That is because we used the same controller gains for both celular robots, even though they have some differences that may be considered, such as wheels width.

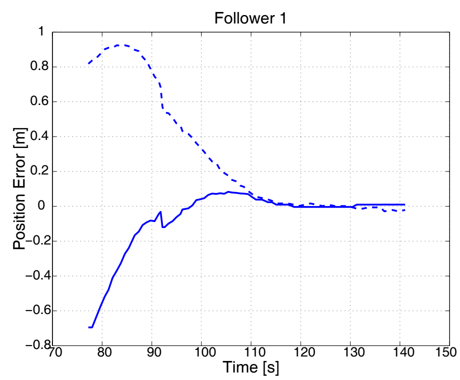


Fig. 9. Position errors for robot-1.

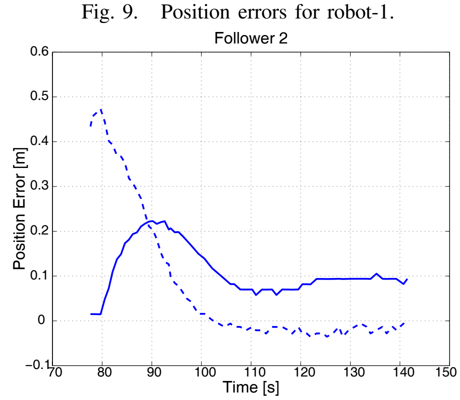


Fig. 10. Position errors for robot-2.

Figure 11 and Figure 12 show the calculated orientations for each follower (solid line) as well as their real performance (dashed line). Although the robots pose measurements are filtered with Kalman filter, there are some oscillations because robot orientation is estimated using just image information.

It is worth to say that the followers orientations must not necessarily approch zero. In this case, their orientation should be 90^o according to the leader reference frame.

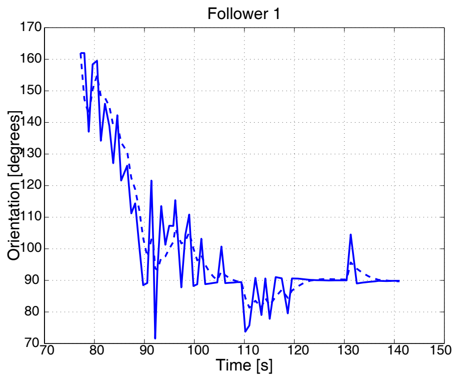


Fig. 11. Robot-1 orientation.

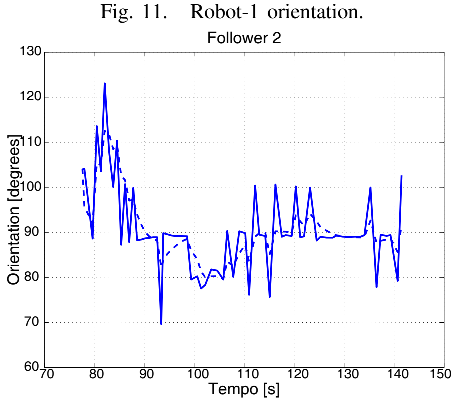


Fig. 12. Robot-2 orientation.

V. Conclusion and Future Work

This work presents a cooperative strategy on mobile robotics based on nonlinear control techniques and omnidirectional vision. The idea is to control a cheap robot team leaded by one robot having the higher processing capacity. The leader is responsible for group navigation and coordination. Group formation (position, orientation and velocity) is accomplished by a stable nonlinear controller with visual feedback. Such controller is the main focus of this paper without concerning obstacle detection and avoidance.

The proposed controller unifies a formation controller, designed to make the team achieve a desired formation, and a compensation controller, which considers the leader motion into the followers velocities. The omnidirectional visual feedback has the advantage of allowing the leader to localize all the followers around itself by catching just one image. By choosing different colors to identify the followers, background subtraction and color segmentation are used for localizing each robot related to the leader. Such visual information is used by the controller to define desired velocities for the followers in order to achieve and keep team formation. Simulations and experiments were carried out to evaluate the controller performance.

Current results are encouraging. For the next steps longer experiments will be executed and the *ZigBee* radio link technology will be used to implement group communication.

Also we shall improve the controller and consider obstacle avoidance to the leader system. 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. Moreover, omnidirectional images may be used to map the environment for a global localization, so the leader will be able to guide the entire group to different places and coordinate tasks execution.

Acknowledgment

The authors would like to thank FAPES, through the project FUNCITEC 30.897.360/2005, 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, 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 new robot team and Celso De La Cruz Casaño (PhD student) for his assistence.

References

[1] M. J. Matáric, M. Nilsson, K.T. Simsarin, Cooperative Multi-Robot Box-Pushing, *IROS 95*, vol. 3, pp. 556-561, 1995.

[2] J. T. Feddema, C. Lewis, D. A. Schoenwald, Decentralized Control of Cooperative Robotic Vehicles: Theory and Application, *IEEE Trans. Robot. Automat.*, vol. 18, no. 5, pp. 852-864, 2002.

[3] D. Rus, B. Donald, J. Jennings, Moving Furniture with Teams of Autonomous Robots, *IROS 1995*, pp. 235-242, 1995

[4] J. P. Desai, J. P. Ostrowski, V. Kumar, Modeling and Control of Formations of Nonholonomic Mobile Robots, *IEEE Trans. Robot. Automat.*, vol. 17, no. 6, pp. 905-908, 2001.

[5] R. C. Kube, H. Zhang, Collective robotics: From social insects to robots, *Adaptive Behavior*, vol. 2, no. 2, pp. 189-218, 1993.

[6] T. Balch and R. C. Arkin, Behavior-based formation control for multi-robot teams, *IEEE Trans. Robot. Automat.*, vol. 14, pp. 926-939, 1998.

[7] S. K. Nayar, Omnidirectional Vision, *8th International Symposium on Robotics Research*, 1997.

[8] S. Baker, S. K. Nayar, A Theory of Single-Viewpoint Catadioptric Image Formation, *Int. Journal of Computer Vision*, vol. 35, no. 2, pp. 1-22, 1999.

[9] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, C. J. Taylor, A Vision-Based Formation Control Framework, *IEEE Trans. Robot. Automat.*, vol. 18, no. 5, pp. 813-825, 2002.

[10] R. Vidal, O. Shakernia, S. Sastry, Following the Flock, *IEEE Robotics & Automation Magazine*, pp. 14-20, 2004.

[11] Z. Zhu, K. D. Rajasekar, E. M. Riseman, A. R. Hanson, Panoramic Virtual Stereo Vision of Cooperative Mobile Robots for Localizing 3D Moving Objects, *IEEE Workshop on Omnidirectional Vision - OMNIVIS'00*, pp. 29-36, 2000.

[12] J. A. Santos-Victor, R. Carelli, S. Van Zwaan, Nonlinear Visual Control of Remote Cellular Robots, *10th Mediterranean Conference on Control and Automation*, 2002.

[13] R. F. Vassallo, L. F. Encarnação, J. A. Santos-Victor, H. J. A. Schneebeli, Bird's Eye View Remapping and Path Following Based on Omnidirectional Vision, *XV CBA*, 2004.

[14] F. G. Pereira, C. C. Gava, R. F. Vassallo, M. Sarcinelli-Filho, Calibração de Sistemas Catadióptricos e Detecção da Pose de Robôs Móveis por Segmentação de Imagens Omnidirecionais, *VII SBAI*, 2005.

[15] Celso De-La-Cruz, R. Carelli, Control Centralizado de Formación Usando Una Cámara Omnidireccional, *IV Jornadas Argentinas de Robótica*, 2006.

[16] R. Carelli, F. Roberti, R. F. Vassallo, T. F. Bastos-Filho, Estrategia de control estable de formación para robots móviles, *AADECA 2006 - XX Congreso Argentino de Control Automático*, 2006.

[17] R. Kelly, R. Carelli, J. M. I. Zannatha, C. Monroy, Control de una Pandilla de Robots Móviles para el Seguimiento de una Constelación de Puntos Objetivo, *VI Congreso Mexicano de Robótica*, 2004.