Control aspects of maintaining non-holonomic robots in geometric formation

António Paulino, Helder Araújo

Abstract. Low-level control is one important element in every robotic system, and affects significantly its performance. Every robotic system needs an adequate control law, which should be chosen having in mind the kind of devices in use, and the type of source of data present in the system. This paper presents two different controllers, and compares their performance, when applied to non-holonomic robots, integrated in formation maintenance systems.

1 Introduction

Maintaining several robots in geometrical formation is useful for different types of applications. There are three main aspects that are very relevant for this type of applications: robot localization, system architecture and low level control. Robot localization can be performed by means of a wide variety of sensors. When vision is used to perform robot localization, it usually employs known landmarks and/or beacons. Commonly, vision results are combined with odometry information from the robots wheel encoders, because of the higher acquisition rate, often using a Kalman filter [Chenavier 92]. Other positional subsystems use devices like laser range finders, sonars or infrared sensors. The advantage of sonars and infrared sensors is the lower computational requirements of both of them.

The architecture of a system of multiple robots defines the relationships among the robots (including the need or not for the robots to communicate among themselves while performing a task). One can have completely independent and anonymous robots, only knowing, in advance, the geometric form to achieve [Sugihara 96][Yamaguchi 97], or a central coordinator, responsible for the definition of the geometric form and the number of elements in the system. In [Paulino00] a mixed architecture is presented, where this central coordinator resides in one of the robots, typically the leader. In some applications (hazardous environments), robots can stop working. Therefore the central coordinator is responsible for informing the remaining robots, so that they can rearrange themselves to cover the same area, with the same geometric form.

Other methods to achieve geometric patterns do not require a leader. In [Sugihara 96], each robot knows all the others positions, and, using that information, it places itself in the formation. In [Wang 1991] and [Yamaguchi 97], each robot relates only to its one or two nearest neighbors.

Finally, the last aspect is the low level control of the robot. This module is responsible for conducting the robot to its position in the formation. In the case where the leader is the main reference for each of the robots, the design of the low level control is basically a trajectory tracking problem.

The study of this low level controller is the main goal of this work. Two different control methods are to be compared: a linear PID controller and a nonlinear controller. Both will be tested in a simulated environment, and their behavior as a function of the parameters related to formation maintenance, will be quantified and compared.

2 Robot pattern formation systems

Several systems for achieving and maintaining geometric patterns between robots have been published, differing mainly in three points: whether or not the robots are individually identified, the communication structure, and the ability to produce different patterns.

In [Balch98], an approach based on the so-called motor schema behaviors is presented. These are high-level actions used to generate the movements needed to achieve a certain goal or formation position. If a goal is composed of a number of behaviors or schemas, as, for an example, gain-formation, avoid-robot and noise, the output is an average of the outputs of the behaviors, weighted by their importance. In this system, each robot is assigned a unique identification number, which will define its position in the formation. A behavior for position acquisition is also defined, which can be accomplished by GPS, TCP/IP link or dead-reckoning, depending on the application.

A rather different system is described in [Sugihara 96] and [Suzuki91], where anonymous robots are used. This means that there is no communication between them. Instead they have a sensor enabling them to monitor the other's positions, in its own coordinate system. Algorithms for the formation of common patterns are presented. These include circles, polygons, line segments and the action of following other robots. The main advantage of this system is the ability to handle a large number of robots, since communication overhead does not exist.

Finally, in [Wang 1991] some navigation strategies are defined based on neighbor tracking. The main idea is to define a deviation vector that robot i has to maintain relative to robot *i-1*. The motion of robot i is then computed from the robot's *i-1* motion. Similar laws are derived for the case of multi-neighbor tracking. In this system each robot needs only to track the nearest robots, as opposed to [Sugihara 96]. However, instability problems can occur, since the formation error propagates from robot *i-1* to robot i.

More specific systems are also found. In [Yamaguchi 97], describing a surveillance application, a group of n robots prevents an intruder to enter an area. It uses a formation vector that each robot has to maintain to close the area. The outer robots have, as reference for stopping, landmarks. In [Ando99] there is a more realistic approach, where each robot has a limited range of perception. However the algorithm describes only a flocking application.

Our system, described in [Paulino00], has a mixed architecture. Each robot is partially independent, and the system can keep running even if one of the robots is lost. The communication only exists with a central server, and is restricted to informing the pattern type and the number of elements in the system. Also a collision avoidance mechanism is incorporated. In respect to the geometric type of pattern, and since the robot positions in the formation are expressed by vectors, the pattern shape is not restricted.

3 Control Laws

For the low level control module of the formation, we need to choose one control law designed for non-holonomic robots. For this problem several methods have already been published.

In the particular case of tracking a leader with an arbitrary path, some of the approaches are not suitable. This is the case of [Kang 99], where a non-time dependent method is developed from a standard control law. The conversion from a time dependent to a time independent law is achieved with a parametric equation of the trajectory to be described, which, in our application, is not known in advance.

The two control laws considered for evaluation are based on common approaches from control theory, a linear and a nonlinear control, which will be briefly described in the next subsections.

Let us assume that all the coordinates are measured in a fixed coordinate frame. The fixed coordinate frame is defined by the initial position and orientation of the follower robot (FR). An important reason for this choice is that FR locates the leader robot (LR) relative to its coordinate frame. A 2D rotation and translation is enough to convert this position to the fixed frame. Hereafter, we will only use three position vectors: $P_F = (x_F, y_F, \theta_F)$, $P_L = (x_L, y_L, \theta_L)$ and $P_d = (x_d, y_d) = (x_L, y_L) - (x_{ref}, y_{ref})_{FR}$, respectively FR position, LR position and the desired position, all defined in FR coordinates. The reference vector $(x_{ref}, y_{ref})_{FR}$ is defined by the position of FR in the formation, as in [Paulino00], and is a function of the LR angle, so the global formation orientation is given by LR.

Notice that the aim of the system is to maintain the relative vector between the two robots, not the relative rotation between robots, which is not possible to achieve simultaneously.

3.1 PID control law

Consider at a certain time instant, P_F, P_L and P_d. We can then compute the error vector $P_E = P_d - P_F$, which defines where to lead the robot, to keep it in formation. Once expressed in polar coordinates l_{P_E} and ρ_{P_E}, called the error length and orientation, as shown in figure 1, the translational and rotational velocities of the FR, respectively v_{TF} and ρ_{TF}, are a function of these error values.

A method to compute these velocities has to be defined. One possible solution is to use a PID controller. In our particular case, the control law becomes:

$$\begin{cases} v_{TF} = \alpha_P . l_{P_E} + \alpha_I . \left(\int l_{P_E} . dt\right) + \alpha_D . \frac{dl_{P_E}}{dt} \\ v_{\rho F} = \beta . \rho_{P_E} \end{cases} \quad (1)$$

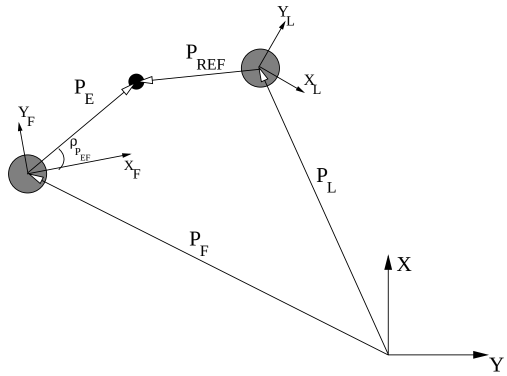

Fig. 1. Vectors used in the computation of the position error of FR

The parameters α_P, α_I, α_D and β are, as will be shown, of extreme importance in the overall system response, as will be studied in the next sections.

The robots use a differential drive, so the translational and rotational velocities must be decomposed to left and right wheel velocities, using 2 from [Crowley 95]:

$$\begin{cases} V_L = v_{TF} - v_{\rho F} . \frac{w}{2} \\ V_R = v_{TF} + v_{\rho F} . \frac{w}{2} \end{cases} \tag{2}$$

where w is the distance between wheels.

3.2 Nonlinear control law

A nonlinear control law defined for the control of multiple robots in formation was presented in [Desai 98]. This control law has the advantage of not requiring the knowledge of the formation trajectory in advance. It can be used to track an arbitrary reference path with a non-holonomic robot. The control law is based on methods of feedback linearization. Feedback linearization techniques algebraically transform a nonlinear systems dynamics into a (fully or partially) linear one.

In [Desai 98], two control laws are defined, $l-\psi$ and $l-l$, both oriented to robot following and formation maintenance. The first control law causes the FR to follow the LR path, while maintaining a desired constant vector $\left(l_{LF}^d, \psi_{LF}^d\right)$.

This law takes as input the actual distance from LR to FR, defined l_{LF}, and the orientation between themselves, defined ψ_{LF}. These values are shown in figure 2. The output are the translation and rotation velocities for FR to maintain a desired vector $\left(l_{LF}^d, \psi_{LF}^d\right)$ between the two robots.

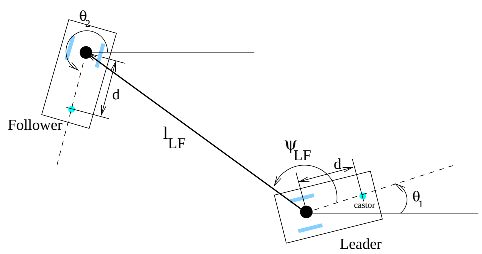

Fig. 2. Vectors used in the nonlinear control law $l-\psi$

As can be seen, in this method, the leader coordinate frame defines the orientation of the formation pattern. In the PID control law case, this orientation can or cannot be defined in this way, depending on whether the reference vector is a function of the LR orientation

Suppose that at a certain time instant the (l_{LF}, ψ_{LF}) vector from LR to FR, and an estimate of the translation and rotation velocities of LR: v_{TL} and $v_{\alpha L}$ are known. In can be proved that, starting from the kinematic equations of a non-holonomic robot, and applying feedback linearization (I/O linearization), we obtain a control law that can be proved to be exponentially convergent. Details of this derivation are published in [Desai 98].

$$\begin{cases} v_{\alpha F} = \frac{\cos\phi}{d}\left[\alpha_2.l_{LF}.\left(\psi^d_{LF} - \psi_{LF}\right) - v_{TL}.\sin\psi_{LF} + \right. \\ \qquad\qquad \left. l_{LF}.v_{\alpha L} + \rho_{LF}.\sin\phi\right] \\ \qquad\qquad v_{TF} = \rho_{LF} - d.v_{\alpha F}.\tan\phi \end{cases} \tag{3}$$

Where: $\rho_{LF} = \frac{\alpha_1.\left(l^d_{LF} - l_{LF}\right) + v_{TL}.\cos\psi_{LF}}{\cos\phi}$ and $\phi = \theta_L + \psi_{LF} - \theta_F$.
Again, the conversion to left and right wheel velocities is done by using equation 2.

4 Comparison of the control laws

Next the two control laws described in section 3 will be compared. As stated before, they will be thoroughly tested in a robot simulator. Certain behaviors, which are most important to the formation maintenance will then be compared. These include gaining formation, following some predefined paths, changing formation and the behavior in presence of measurement noise.

An important point studied here is the effect of the control parameters in the performance of the system. In the case of the PID controller, we have three parameters: proportional, integral and derivative, while for the nonlinear law we have only two. To have a more accurate comparison of the laws, we tested all the situations with different groups of parameters. We started with a group with average behavior, which we found to be $P = 0.009, I = 0.5, D = 0.5$ for the PID and $\alpha_1 = 0.1$, $\alpha_2 = 0.3$ for the nonlinear control law. Each of the parameters was then changed, enabling an evaluation of its effect in the performance of the system.

These average behavior groups of parameters were found from tests to the system. They showed not to be too reactive, nor too sluggish, giving then a good starting point to the performance evaluation.

One remark should be made: the intention of this paper is to perform an experimental comparison of the two methods.

4.1 Simulation environment

The environment chosen for the simulations was the robot simulator from Nomadic Technologies, which is very reliable and straightforward to use.

Each robot is controlled by an independent process, following the topology described in [Paulino00].

In essence, it is formed by a central server, to which all clients connect. In the client-server direction the client's world position is transmitted, and in the opposite direction, the client receives the neighbors positions, measured in its own coordinate system, which will be used by the control module. The coordinate transformation is done in the server.

In fact, if a real environment implementation uses vision as the main sensor, the neighbor positions are measured in camera coordinates, and therefore in robot coordinates. The simulation architecture implements this behavior, so that the control subsystem can be tested as if it was a real system.

This simulation environment, as described above, can lead to delay problems. This delay is the time occurring between the transmission of the absolute position from client to server, and the reception, by the client, of the relative positions of the neighbors. It is due to the processing time and the TCP link. Nevertheless, our tests showed that it only affects the performance when the robot position changes abruptly. Since we limited the velocities (as a real robot has velocity limitation), the delay problem does not affect the tests.

4.2 Gaining formation

In this test, we initialize FR at position $(500, 0)$ and LR at $(0, 0)$. The first must then take its place in the formation, which we define to be at $(1000, 180^o)$, relative to LR. Throughout these tests, the distances are measured in centimeters, and the angles in degrees.

Applying the PID controller, we get the results presented in the graphics of figure 3. The figure at left represents the evolution of the position of LR and FR in the internal XY map of the FR, while the picture at

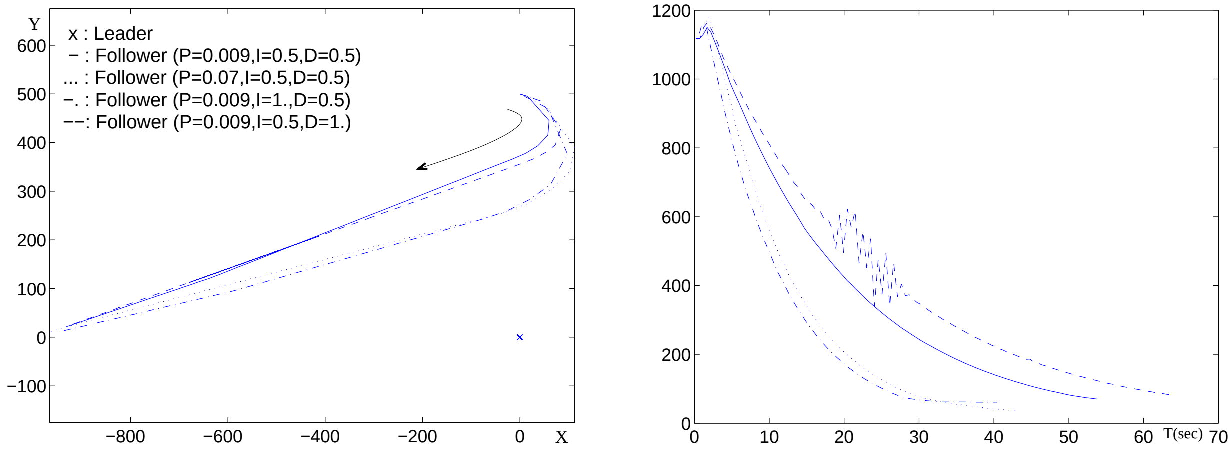

Fig. 3. Results for the gaining formation test, using the PID controller

right denotes the variation of the length of the error as a function of time. Each line represents an execution, with different parameters.

As it can be seen, the choice of the PID parameters significantly affects the behavior of the system, and the influence of each parameter is perfectly clear in figure 3. As expected, a high derivative coefficient can lead to instability on the FR response, whereas an increase on the integral and/or the proportional coefficients causes the convergence to be quicker.

In figure 4 are shown the same tests, now using the nonlinear control law.

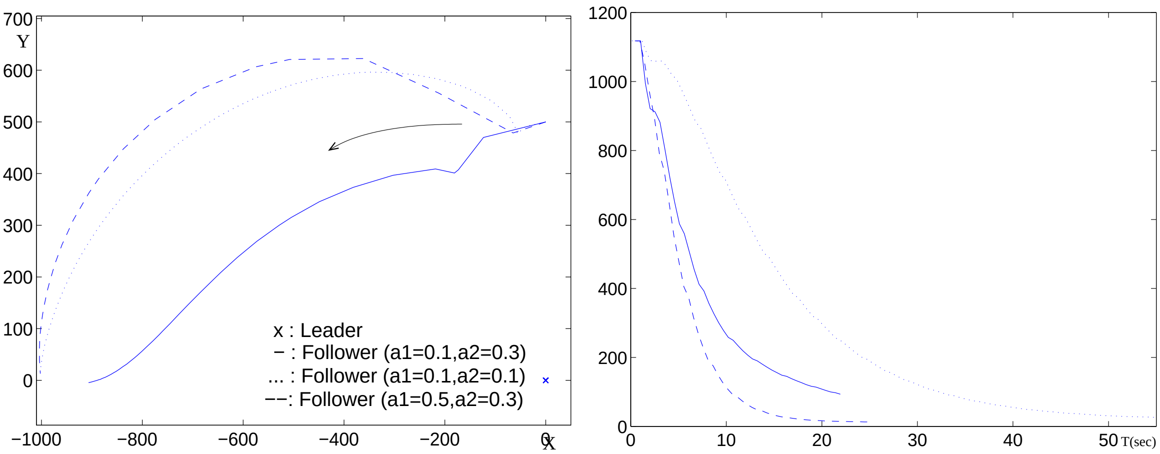

Fig. 4. Results for gaining formation, using the nonlinear controller

It is clear that, generally, the convergence is much faster than using a PID. The only problem noticed with this control law was that, if the initial position is very far from the desired point, the system easily becomes unstable, unless the control parameters are set to lower values, which causes the convergence to be slower.

This instability does not appear in the PID controller, for a large range of parameters.

Figure 5 measures the influence of the distance and parameter α_2 in the stability.

The criterium is the average distance between the trajectory described, using as parameters $(dist, \alpha_2)$, and the optimal trajectory, which is a straight line between the start and final position. Note that $dist$ is the euclidean distance between the initial and final configuration.

It is clear that for a combination of large distance and large parameters, the convergence is worse, causing the system not to converge.

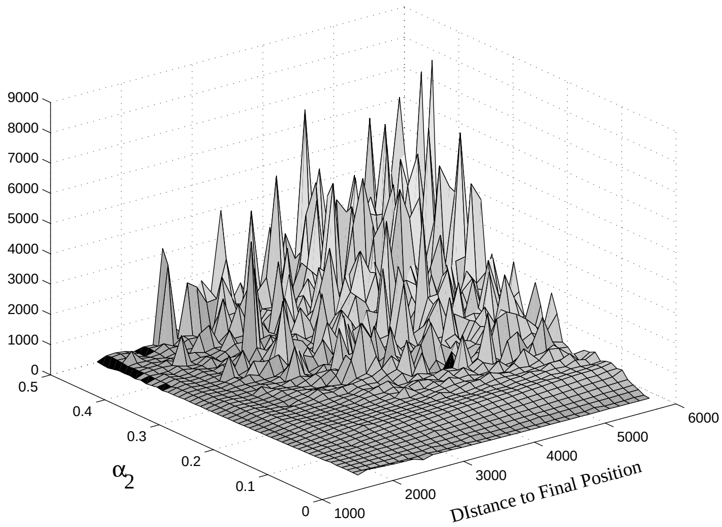

Fig. 5. Performance of the nonlinear law, for various configurations and parameters

4.3 Path following

The next test is, as stated above, path following. After we have FR in its position in the formation, we apply a trajectory to the LR, which FR should follow.

The first trajectory is the simplest one, a straight line. Having FR and LR both stopped, we set LR velocity to some value. The resulting instantaneous error length are shown in figure 6, with the PID at left and the nonlinear at right.

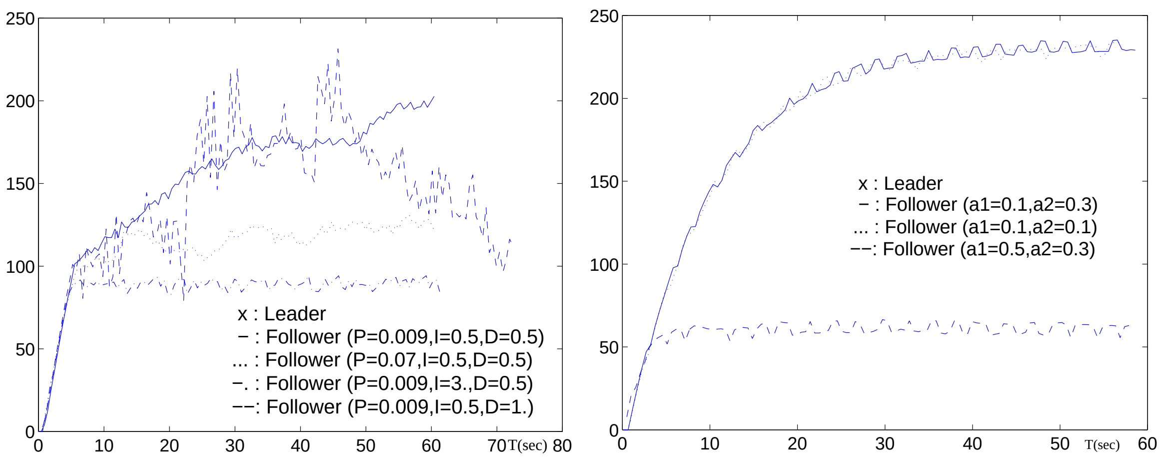

Fig. 6. Error lengths for the path following of a line

From figure 6 it is obvious that the PID controller leads to more oscillations than the nonlinear controller. As a matter of fact the PID controller is more difficult to tune: an increase in the proportional coefficient leads to a smaller error, but also to a less stable behavior. The integral coefficient is also important, since it reduces the steady-state error.

In the case of the nonlinear control law, the tracking of a straight line is only affected by α_1, since, from eq. 3 is directly related to the translation velocity, which does not happen with α_2. In figure 6 we notice the particularly low steady-state error, after increasing α_1. This can be an advantage over the PID, but leads to a stability problem, as mentioned in section 4.2.

In real situations were LR leads FR, its path is not restricted to a straight line. Since we cannot test the performance for every path, we defined a more complex one, composed of a straight line, a turn to the left, another straight line, with higher velocity, another turn the right and a line. This path is shown in figures 7 and 8.

These figures also show the results of following the path. Notice that the coordinate system of the leader defines the orientation of the formation. This implies that, when it turns, he reference vector the FR follows also turns, and the error becomes, for a while, very large. This is clear from the position graphics of FR.

The time that the FR takes to recover is an important evaluation of the controller. So, in figures 7 and 8 it can be seen that the nonlinear controller recovers more rapidly and with less oscillation than the PID.

It should also be noticed the lower error lengths with the PID. This is caused by a faster response to an increase on the error.

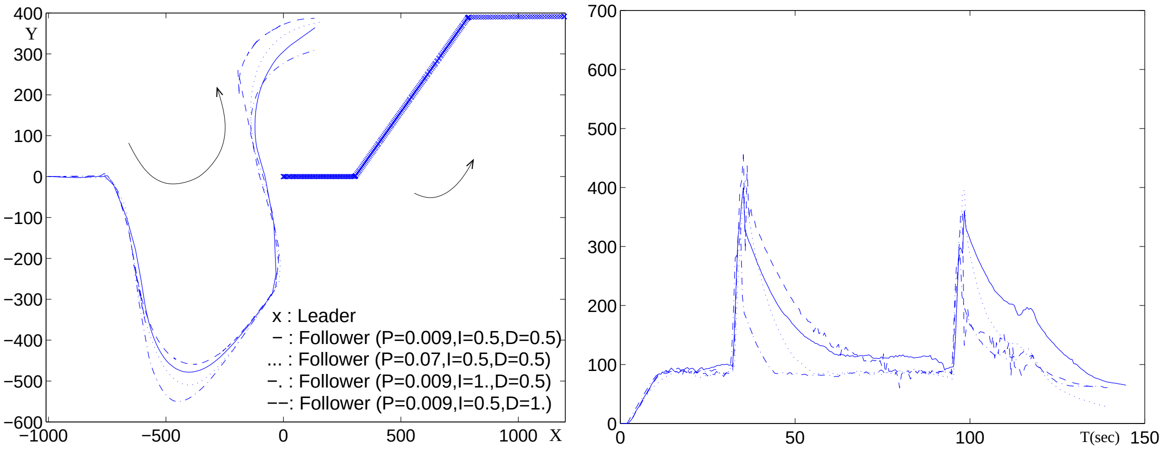

Fig. 7. Behavior of different PID parameters following an arbitrary path

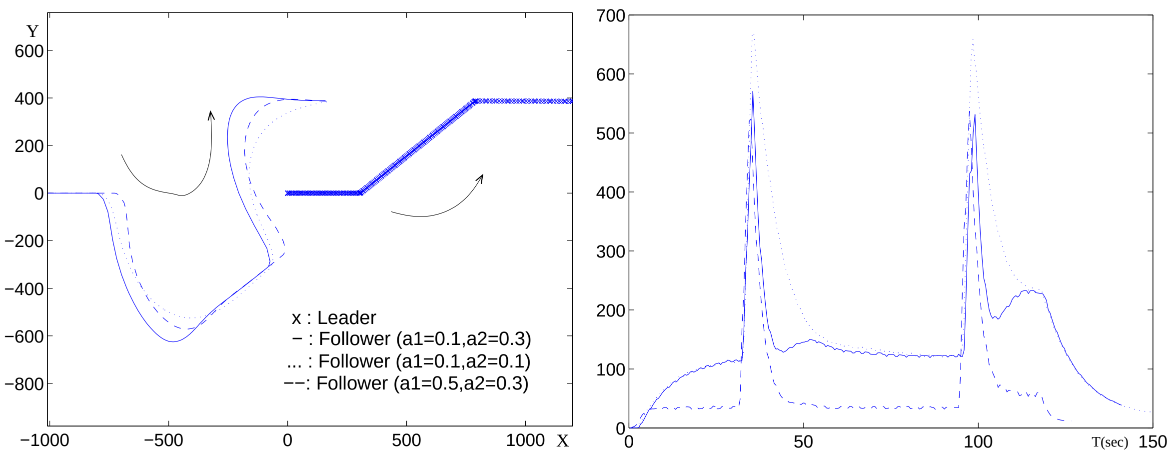

Fig. 8. Behavior of different nonlinear control parameters following an arbitrary path

4.4 Formation change

In the current system defined in [Paulino00] formation can be changed dynamically. This means that the formation can change from a line to a circle (for example). This basically amounts to a change in the reference vector. The response to such a sudden change was also studied, and is represented in figures 9 and 10.

In the first part, the reference vector was as defined in section 4.2, and, having FR following LR in a straight line, the reference was changed to $(500, 90^o)$.

Figures 9 and 10 confirm the results from the previous section: the response of the nonlinear law is typically much faster than PID's.

It also becomes clear that the PID response leads to more oscillations than the nonlinear.

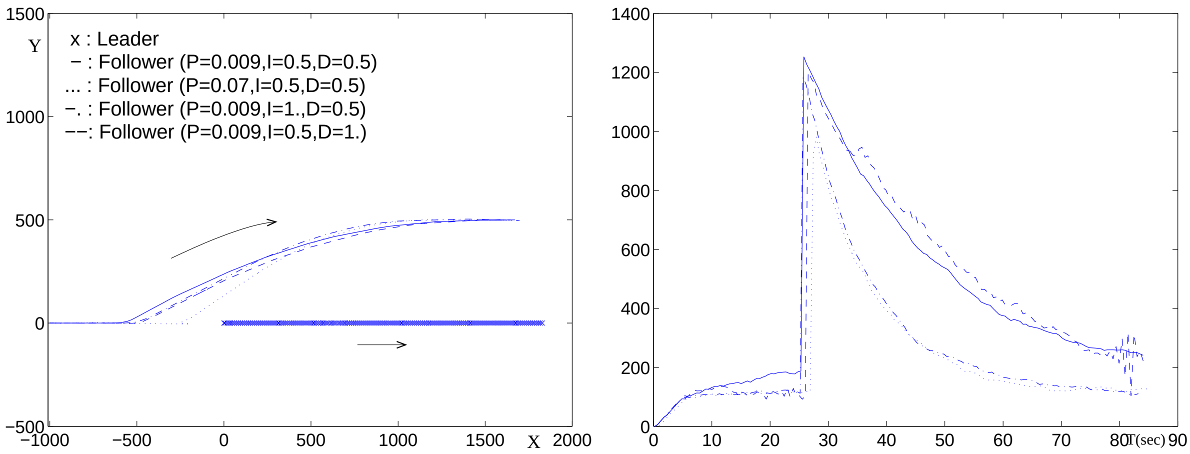

Fig. 9. Formation change results for the PID controller

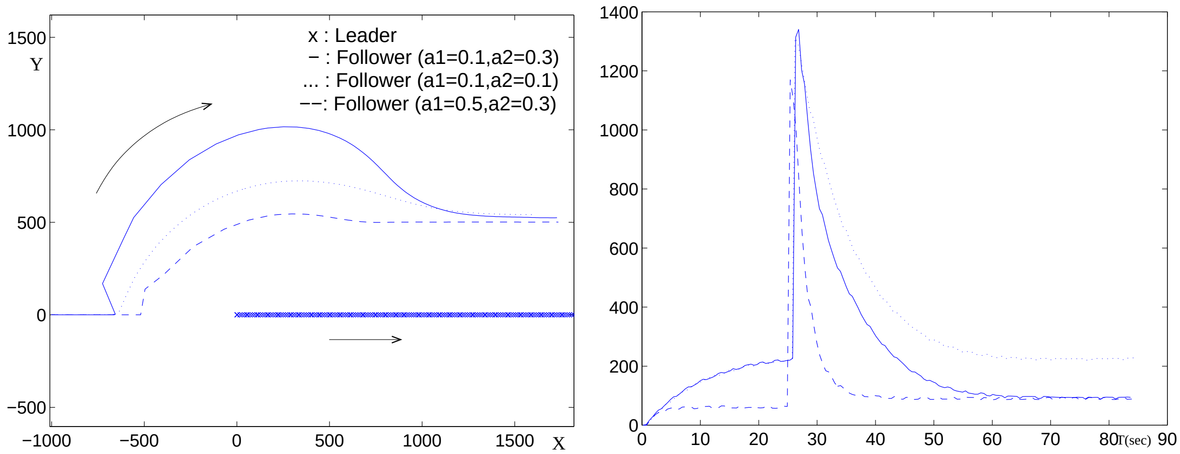

Fig. 10. Formation change results for the nonlinear controller

4.5 Influence of noise

After testing all the most important situations in geometric formation maintenance between robots, it is important to consider the effects of noise, since measurements are always affected by noise. Therefore it is important to evaluate the performance of the control law in the presence of noise.

This test was done using the arbitrary path defined in section 4.3, with a contamination of the leader position with additive gaussian white noise (zero mean). Different variance noise values were considered, yielding the results presented in figure 11. In this figure, TV refers to the maximum error introduced in the x and y coordinated of LR, while RV corresponds to the error introduced in θ of LR.

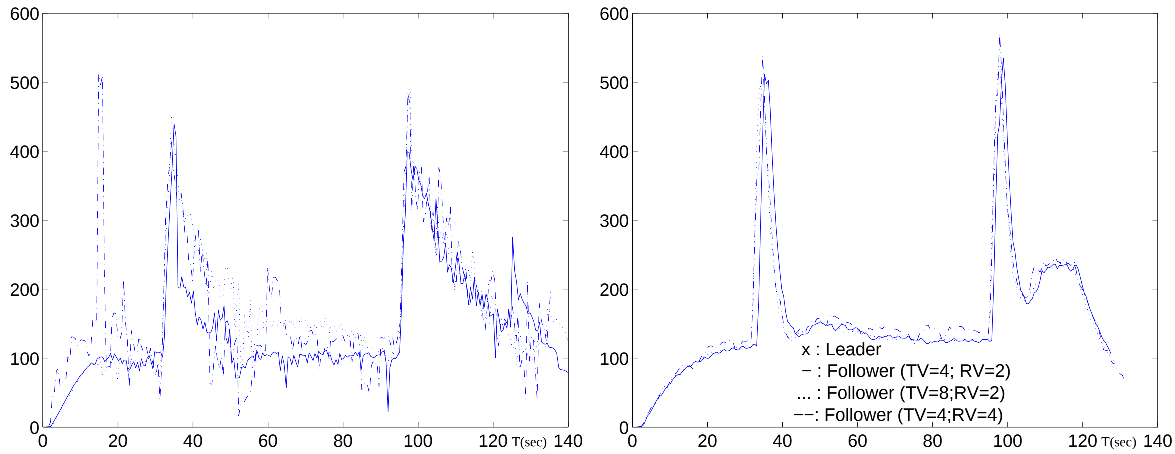

Fig. 11. Response of the PID controller and the nonlinear controller in the presence of measurement noise

5 Conclusion

As an important conclusion it was verified that the nonlinear control law proved to be more stable, whereas the PID controller lead to responses with many oscillations.

One important remark regarding the results obtained by the nonlinear controller, is that its equations make use of rotation and translation velocities measured by the robot simulator, which are more accurate than the equivalent derivatives computed by the PID control law. In the case where noise is present, this can be particularly important.

Nevertheless, the use of nonlinear control law in real systems is very limited when the robots are independent. In fact, if the FR is estimating the information about the LR, it can be a problem to estimate its velocities. In our system, vision is being used, and if the position information is estimated with its inherent errors, the corresponding estimates of the velocities can not be obtained, in general, with acceptable accuracy. Because of this, only the PID was implemented. A new technique of estimating velocities from image is being developed, in order to have the nonlinear control implemented in a real system.

In our real system, we verified that the results presented in the last section hold. In spite of that, with a careful tuning, and the use of a so-called “ballistic zone”, the oscillations can be minimized. However, should another sensor be used, enabling the computation of rotational an translational velocities, and the nonlinear control law should be employed.

6 Acknowledgements

The authors gratefully acknowledge the support of project PRAXIS/2/2.1/TPAR/2074/95, funded by the Portuguese Foundation for Science and Technology. António Paulino also acknowledges the support of project PRAXIS/P/EEI/10252/1998, funded also by the Portuguese Foundation for Science and Technology.

References

[Chenavier 92] F. Chenavier, J. Crowley; Position Estimation for a Mobile Robot Using Vision and Odometry; Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France, May 1992

[Paulino00] A. Paulino, J. Batista, H. Araujo; Multile Robots in Geometric Formation: Control Structure and Sensing; Proceedings of the 2000 International Symposium on Intelligent Robotic Systems; Reading, England, June 2000

[Sugihara 96] K. Sugihara, I. Suzuki; Distributed Algorithms for Formation of Geometric Patterns with Many Mobile Robots; Journal of Robotic Systems 13 (3), 127-139, 1996

[Wang 1991] P. K. C. Wang; Navigation Strategies for Multiple Robots Moving in Formation; Journal of Robotic Systems 8 (2), 177-195, 1991

[Yamaguchi 97] H. Yamaguchi; Adaptative Formation Control for Distributed Autonomous Mobile Robot Groups; Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997

[Kang 99] W. Kang, N Xi; Non-time referenced Tracking Control with Application in Unmanned Vehicle; IFAC, 14th Triennial World Congress, Beijing, P. R. China, 1999

[Crowley 95] J. Crowley; Mathematical Foundations of Navigation and Perception for an Autonomous Mobile Robot; Workshop on Reasoning with Uncertainty in Robotics, University of Amsterdam, The Netherlands, December 1995

[Desai 98] J. Desai, V. Kumar; Controlling Formations of Multiple Mobile Robot; Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 1998

[Balch98] T. Balch, R. C. Arkin; Behaviour-based Formation Control for Multi-robot Teams; IEEE Trans. on Robotics and Automation, Vol 14, N.6, December, 1998

[Suzuki91] I. Suzuki, M. Yamashita; Distributed Anonymous Mobile Robots - Formation of Geometric Pattern; SLAM I. Comp, Vol. 28, N.4, 1991, pp. 1347 - 1363

[Ando99] H. Ando, Y. Oasa, I. Suzuki; Distributed Memoryless Point Convergence Algorithm for Mobile Robots with Limited Visibility; IEEE Transactions on Robotics and Automation, Vol. 13, N.5, October 1999

This article was processed using
the TEX macro package with SIRS2001 style