A Generic Framework for Multi-robot Formation Control

Tiago P. Nascimento[1,*], André Gustavo S. Conceição[2],
Hugo P. Alves[1], Fernando A. Fontes[1], and António Paulo Moreira[1]

[1] FEUP - INESC Porto, University of Porto,
4200-465 Porto, Portugal
`tiagopn@ieee.org`,
`{hugo.alves,faf,amoreira}@fe.up.pt`
[2] Federal University of Bahia - Brazil,
40210-630 Salvador, Bahia, Brazil
`andre.gustavo@ufba.br`

Abstract. This paper describes a novel approach in formation control for mobile robots. A Nonlinear Model Predictive Controller (NMPC) is used to control the formation of a heterogeneous mobile robots group. The desired formation is formed by an holonomic robot and a nonholonomic robot. The same nonlinear controller is used in both robots with the same cost function. The details of the controller structure are presented in order to track a fixed target departing from different positions in the field avoiding collisions with each other. A soccer robot competition field is used to present the simulations to evaluate the performance of the controller.

Keywords: Formation Control, Nonlinear Model Predictive Controller, Mobile Robots.

1 Introduction

An adaptive framework based in predictive control for creation and maintaining of a mobile robot team formation was conceived as main objective of this work. A formation is usually defined as the special arrangement of a set of agents of the same type, where the relative positions of its elements are steady even if the formation is moving. The used formation differs from the usual rigid formations where the relative position of a team element must be precisely maintained. Here, the ideal formations are the ones that maximize the team perception of the environment or of an element that can be a leader robot or of a moving target.

The three major approaches used in multi-robot formation are: Virtual Structure, Behavior-Based and Leader-Following. This last one being one of the most

studied in multi-robot formation [3], [7] and [4]. Nevertheless, it's important to mention that different techniques forming the Decentralized approach have always been sustained using artificial potential fields [1], constrained force [14] or path planner strategies [9]. Nevertheless, a good review of the three major approaches in formation control can be seen in [2]. In this paper, the leader-following approach will be used. The leader here will be the target (in this case the ball). Nevertheless, here the target is fixed as the objective is the convergence of the formation.

One of the most used controllers in the leader-follower approach is the Model Predictive Controller. It has been the target of study in multi-robot motion control in almost a decade [5]. In 2008, the first use of a MPC applied in the leader-following approach using holonomic robots was done by [8]. In these approaches only the formation maintenance is discussed. In these works the circle trajectory and the eight trajectory were used. The robots should follow the paths in a pre-set time while changing their formation (column or triangle). It's known that in a highly dynamic environment, if the trajectory is pre-defined, the linear MPC, even though applied to a non-linear system, can control the system maintaining the set-point. This was the exact result given by the authors.

In the following year, [6] applied a Linear Model Predictive Controller in the leader-following approach. They applied it in nonholonomic robots using the separation principle to make a NMPC control the trajectory while a MPC would be used to formation control. Both works only had simulation results and no obstacles were considered.

Finally, departing from the idea of using the separation principle done by [6], this paper presents a generic Nonlinear Model Predictive Controller (NMPC) framework to converge the formation using a holonomic and a nonholonomic robot around a tracking target.

2   Formation Control

The controller used in this work to formation control was a Non-linear Model Predictive Controller (NMPC). The general structure of this controller can be classified in three types: distributed, centralized, or hybrid. These categories are based on the way the control signals of each robot are calculated.

Here, the distributed architecture was chosen as can be seen in Fig. 1. In this case, each one of the robots calculates the total control inputs U_n solving its own optimization problem. This takes away the dependency from a central processing unit, guaranteeing the functioning of the formation even in cases of communication failure. Therefore, each robot must have information about the state X_n (position and speed) of each mate of its team. Also, in case of the communication failure or supervisor failure, the robot uses its predicted open-loop strategy to determine these informations, having, therefore, a tolerance degree to failure. Nevertheless, it has the disadvantage of putting a cost in computing the simulation of the entire formation progression, which is done by each one of the robots. However, this was not a problem, for the robots only calculate their own
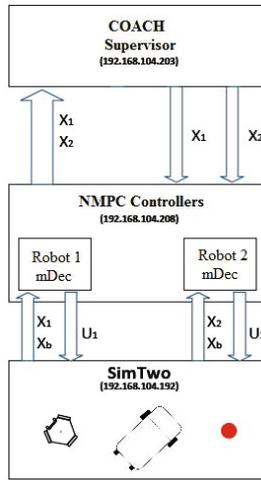
Fig. 1. Distributed Architecture of NMPC Controller

control inputs. As each robot solves its optimization problem in a decentralized architecture, the formation becomes difficult to stabilize.

In this multi-robot formation control case of study, a simulation software called *SimTwo* was used to simulate the formation [10]. In this simulation, the *SimTwo* has the job of another software called HAL (Hardware Abstraction Layer), which is an application that receives the sensor signals and communicates with the actuators, and then with the mDec (software of control of the real robots) by UDP protocol. In the real robots, the HAL sends to the robot's mDec the state of the other robots and the state of the ball. Then, each mDec sends to the *SimTwo* the control references of its robot. Each mDec also communicates with another central computer (the supervisor) that contains the Coach software, sending its own state and the state of the ball while observing it. Finally, the Coach sends to each mDec individually the state of the other robots in formation, in a way that each robot has the information of position and velocity of its mates. It can be noticed that this arrangement is similar the one used in real experiments, where the only difference is the replacement of the *SimTwo* for the HAL in each robot.

The capacity of the NMPC controller to create and maintain a formation comes from the fact that cost functions used by the controllers of each robot in the team formation are coupled. This coupling is done while the information about the position and speed of the other robots are used in the cost function of each robot to penalize the geometry or desired objective deviation. This turns the entire group formation stable where the actions of each robot affect the other mates. Fig. 2 exposes the structure of the used controller. This controller can be divided in three parts:
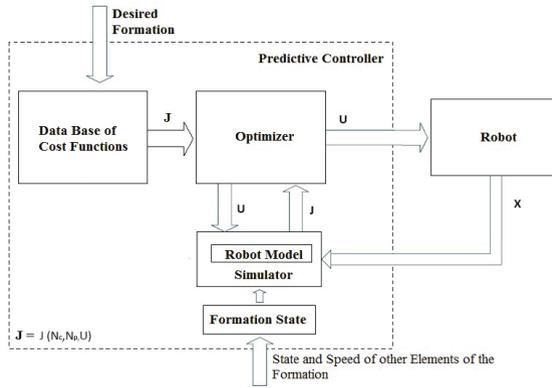
Fig. 2. Structure of the Formation Controller Projected

- **State of the Formation** - The controller contains structures to keep the formations state (position and speed of each other robot in the formation or of any target that should be followed), updating them in each control loop. These informations can be received by a supervisor or by other robots from the team, or even by the robot itself using its own resources;
- **Optimizer** - This part uses a numeric minimization method to optimize the cost function and obtain the signals of optimal control. Here it is used a method called Resilient Propagation (RPROP), which guaranties quick convergence;
- **Simulator** - This part does the simulation not only of the robot state evolution but also the state evolution of the other elements in the formation (other robots or targets). This element uses a dynamic simplified model to emulate the robot evolution. The speeds of the other robots or targets are assumed during the entire horizon of prediction as being constant and equals to the actual speed.
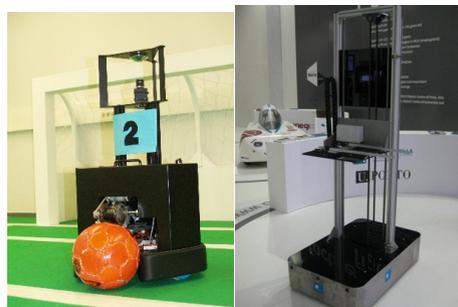

Fig. 3. The used holonomic robot (left) and nonholonomic robot (right)

The Resilient Propagation algorithm (RPROP) appeared in the learning algorithms category used in neural networks [11], being adapted to this application. This is an adaptive method where the step value is not proportional to the gradient function value to be minimized in a desired point (as it happens in the Steepest Descent algorithm), but it keeps adapting with the function behavior. Therefore, it becomes immune to the uncertainties of the derivative function value, depending only on the temporal behavior of its signal. This algorithm was tested initially with the values suggested by [11] and it revels to be capable to converge where the Steepest Descent failed.

3   Problem Formulation

The developed framework was applied to a formation with one holonomic mobile robot from the FEUP's 5DPO team and one nonholonomic mobile robot also from FEUP (Fig. 3). These robots can fulfill one main objective: the optimization of the target relative state perception (ball relative state perception) using the nonholonomic robot as observer while the holonomic robot places itself in an ideal position to receive the ball (receiver). The robots should converge to this formation departing from different positions and avoiding the collisions between themselves or with the target.

The mathematical definition of the system can be understood as having two robots and a ball (target). One of the robots is an omnidirectional robot composed by three holonomic wheels and the second robot is a differential robot with two normal wheels aligned in the center of the robot. Taking as base for this formation definition the elements presented in Fig. 4. The ball position and speed vectors in global coordinates are respectively:

$$X_{ball}(k) = \begin{bmatrix} x_{ball}(k) & y_{ball}(k) \end{bmatrix}^T, \tag{1}$$

$$v_{ball}(k) = \begin{bmatrix} vx_{ball}(k) & vy_{ball}(k) \end{bmatrix}^T. \tag{2}$$

It is considered also that the unit vector of the ball's velocity

$$\hat{v}_{ball}(k) = [\hat{vx}_{ball}(k), \hat{vy}_{ball}(k)]^T. \tag{3}$$

is such that:

$$\hat{v}_{ball}(k) = \frac{v_{ball}(k)}{\sqrt{vx_{ball}^2(k) + vy_{ball}^2(k)}}. \tag{4}$$

For each robot n, its state is represented by:

$$X_n(k) = \begin{bmatrix} x_n(k) & y_n(k) & \theta_n(k) \end{bmatrix}^T, \tag{5}$$

$$V_n(k) = \begin{bmatrix} vx_n(k) & vy_n(k) & w_n(k) \end{bmatrix}^T. \tag{6}$$
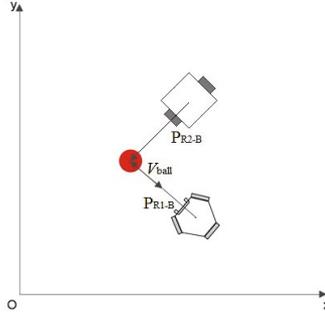
Fig. 4. The desired formation

Note that for the nonholonomic robot, $vy_n(k) = 0$. The position of the ball with respect to robot n is given by $P_{Rn-B}(k) = [x_{Rn-B}(k), y_{Rn-B}(k)]$, where:

$$P_{Rn-B}(k) = \left[(x_{ball}(k) - x_n(k)) \, (y_{ball}(k) - y_n(k)) \right]. \tag{7}$$

Then, it shall be defined the unit vector $\hat{P}_{Rn-B}(k) = [\hat{x}_{Rn-B}(k), \hat{y}_{Rn-B}(k)]$, which indicates the direction of the ball with respect to the robot, and its angle θ_{Rb-B}:

$$\hat{P}_{Rn-B}(k) = \frac{P_{Rn-B}(k)}{\sqrt{x_{Rn-B}^2(k) + y_{Rn-B}^2(k)}}, \tag{8}$$

$$\theta_{Rb-B}(k) = atan2(y_{Rn-B}(k), x_{Rn-B}(k)). \tag{9}$$

Finally, there is also the definition of the positions of each robot n with respect to its mates y, given by $P_{Rn-Ry}(k) = [x_{Rn-Ry}(k), y_{Rn-Ry}(k)]$, where:

$$P_{Rn-Ry}(k) = \left[(x_n(k) - x_y(k)) \, (y_n(k) - y_y(k)) \right]. \tag{10}$$

3.1   The Observer Robot

The estimation of the quality of the ball state is a function of its moving direction with respect to the robot and the distance in between. This estimation is done by using an omnidirectional vision system. Therefore, it's clear that in this case, the robot's direction is irrelevant. If the ball is in movement, the robot should be as indicated in Fig. 4. As in this case the ball is not moving, the robot's orientation becomes also irrelevant.

Nevertheless, big distances between the robot and the ball results in failure of the ball's detection. Consequently, this leads to the failure to estimate its velocity. When the distance is too small, it can occur that the robot cannot see the entire ball and, therefore, become incapable to detect correctly its position increasing the risk of undesired collisions. Take as an example the case in which the distance between the ball and the robot decreases with time in a straight

line. In this case the robot only sees the ball increasing in size, making it difficult to estimate its velocity. In the ideal case the ball should move perpendicular to its position with respect to the robot. Therefore, the desired formation for the observer robots to be around the ball in a way to better estimate the ball velocity possesses the following characteristics:

- The observer robot puts itself by the side of the ball, maintaining a parallel velocity with respect to the ball, v_{ball}, with the same modulus. In this case, as the velocity of the target is zero, the position around the target is irrelevant;
- The robot position vector with respect to the ball, P_{Rx_B}, must be perpendicular to the ball's velocity vector, v_{ball}. Again, in this case this product is null for the target is fixed;
- The robot must maintain a distance $|P_{Rx_B}|$ from the ball;
- The robot must not collide between them or with the target.

Therefore, taking into account all the elements previously described, the weights given to each one of them, and a penalization term to the variation of control effort, the cost function that represents all this, embedded in the observer robot is as follows:

$$
\begin{aligned}
J(N_1, N_2, N_c) = & \sum_{i=N_1}^{N_2} \lambda_1 (d_{setpoint} - |P_{Rn-B}(i)|)^2 + \\
& \sum_{i=N_1}^{N_2} \lambda_2 (\hat{P}_{Rn-B}(i) \cdot \hat{v}_{ball}(i))^2 + \\
& \sum_{i=N_1}^{N_2} \lambda_3 ((\frac{1}{-d_{min} + |P_{Rn-Rm_1}(i)|})^2 + \\
& (\frac{1}{-d_{min} + |P_{Rn-Rm_2}(i)|})^2) + \\
& \sum_{i=1}^{N_c} \lambda_4 (\Delta U(i))^2,
\end{aligned}
\tag{11}
$$

Where N_1, N_2 is the prediction horizon limits, in discrete time, so that $N_1 > 0$ e $N_2 \leq N_p$, where N_p is the desired prediction horizon. Also, N_c is the control horizon, λ_1, λ_2, λ_3, λ_4 are weights for each component of the cost function and $\Delta U(k) = [v_r(k) - v_r(k-1)] + [vn_r(k) - vn_r(k-1)] + [w_r(k) - w_r(k-1)]$ is the variation of the control signals, with $U(i)$ being the reference velocities vector with respect of the center of mass of the robot.

3.2   The Receiver Robot

The ideal position of the receiver robot with respect to the ball to have a good reception of it corresponds to the one in which the robot velocity vector is

collinear with the ball velocity vector, with the same modulus. Also, the robot orientation should be such that the front of the robot is turn towards the ball. Therefore, the robot can then slowly decelerated and the distance between it and the ball can be decreased in a way to receive the ball in ideal conditions.

Summarizing it, the formation here should possess the following characteristics:

- The robot's velocity has to be equal in modulus and direction to the ball's velocity v_{ball};
- The robot's position vector with respect to the ball, P_{Rn-B}, must be collinear to the ball's velocity vector, v_{ball};
- The robot's orientation θ_n must be at all times equal to the vector P_{Rn-B}'s angle, defined by θ_{Rn-B}, in a way that the kicker of the robot is always turn towards the ball;
- The robot must be at a distance $|P_{R-B}|$ from the ball.

Finally, joining all the elements previously described, the weights given to each one of them, and a penalization term to the variation of control effort, the cost function that represents all this, embedded in the receiver robot is as follows:

$$
\begin{aligned}
J(N_1, N_2, N_c) = & \sum_{i=N_1}^{N_2} \lambda_1 (d_{setpoint} - |P_{Rn-B}(i)|)^2 + \\
& \sum_{i=N_1}^{N_2} \lambda_2 (-1)^2 + \\
& \sum_{i=N_1}^{N_2} \lambda_3 ((\frac{1}{-d_{min} + |P_{Rn-Rm_1}(i)|})^2 + \\
& (\frac{1}{-d_{min} + |P_{Rn-Rm_2}(i)|})^2) + \\
& \sum_{i=N_1}^{N_2} \lambda_4 (diff Angle(\theta_n, \theta_{Rn-B}))^2 + \\
& \sum_{i=1}^{N_c} \lambda_5 (\Delta U(i))^2,
\end{aligned}
\tag{12}
$$

Where N_1, N_2 is the prediction horizon limits, in discreet time, so that $N_1 > 0$ e $N_2 \leq N_p$, where N_p is the desired prediction horizon. Also, N_c is the control horizon, λ_1, λ_2, λ_3, λ_4, λ_5 are weights for each component of the cost function and $\Delta U(k) = [v_r(k) - v_r(k-1)] + [vn_r(k) - vn_r(k-1)] + [w_r(k) - w_r(k-1)]$ is the variation of the control signals, with $U(i)$ being the reference velocities vector with respect of the center of mass of the robot.

4   Results

Once the formation algorithm was implemented, some simulations were made to validate the proposed controller and to test its performance under different conditions.

There are many variables that influences the quality of the result. Among them are the weights (the λ_i) of each cost function and the optimizer parameters. The cost function values for both observer and receiver robots can be seen in table 1. In the minimization of the cost function values, only the relationship between the weights given to each element that meters to the final result. Therefore, the final values were a result of an iterative process where the weights are slightly different from one robot to another due to the existing physical differences. This process did not need to be very precise, due to the fact that there were a very large range of weights that could give similar results. Nevertheless, the NMPC controller parameters were $N_p = 10$, $N_c = 2$ and the used reference trajectory to find them was an gate signal extracted in a previous work done by [12].

Table 1. Weights for the Observers and Receiver

Weight	Observer Value	Receiver Value
λ_1	30	5
λ_2	10	10
λ_3	100	100
λ_4	10	5
λ_5	-	10

The initial parameters used on the RPROP optimization algorithm were the ones suggested by [11] ($\eta^+ = 1.5$, $\eta^- = 0.5$, $\Delta_0 = 0.1$) where the algorithm description can also be found. The fist tests resulted in a very satisfactory performance by the controller. After 20 interactions, some changes made on these values were tested ($\eta^+ = 1.2$, $\eta^- = 0.8$, $\Delta_0 = 0.05$) and produced visible improvements.

Therefore, the following simulation results made with the formation control framework evaluate the proposed controller. The simulations for formation convergence are shown to evaluate the formation controller in the following subsection.

4.1   Formation Convergence Results

The following results show the trajectories followed for each one of the robots when, starting from different positions, converge to a preset formation. The target in these cases is stationary during the simulation making the internal product of any vector with the ball velocity vector equals to zero. It is important to notice that the robot number 2 is the observer robot (the nonholonomic AGV)

which means that it can be placed in any position exactly 2m far from the ball, while the robot number 1 is the receiver robot (the holonomic 5DPO) which means that it must be placed in front of th ball. The desired distance between both robot to the ball was defined to be 2 m.

Simulation 1. In this simulation the robots start at positions perfectly opposites and far from the ball (Robot 1 in position (-7,0) and Robot 2 in position (7,0)). For having less risk of collision or probability of the robots to interfere with each other, this became the simplest case. The results can be seen in Fig. 5. From this simulation on it can be noticed some interaction between the robots. The robot 1 goes to the front of the ball while the robot 2 places itself anywhere at the desired distance. As the robot 2 arrives first and places itself in front of the target, the robot 1 stops near the robot 2 avoiding the collision and placing itself as close as possible to the desired position (front of the ball) at the desired distance. The convergence is made in 15 seconds due to the attempt of robot number 1 to position itself in front of the target.
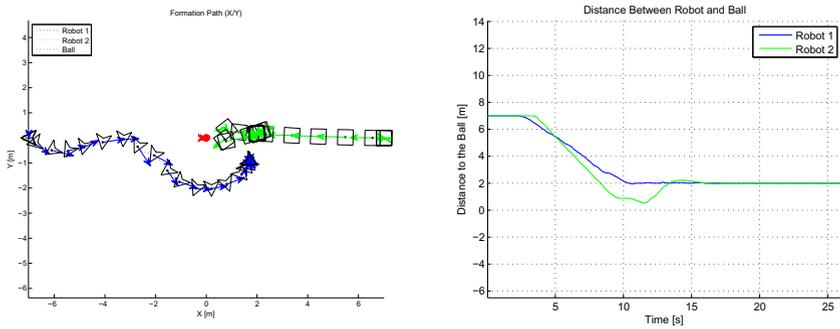

Fig. 5. Convergence into formation, simulation 1

Simulation 2. Here all robots start from the same side of the ball (Robot 1 in position (-7,4) and Robot 2 in position (-7,-4)), thought separated by a distance of 8 m. The results can be seen in Fig. 6. The robots converge perfectly to their positions in formation, making simple trajectories towards the target. As can be seen in the plot of the distance with respect to the time, it can be estimate that the robots have converged to the desired formation in approximately ten seconds.

Simulation 3. The third simulation shows a more complex situation, where the robots start from the same alignment with respect to the ball (Robot 1 in position (-7,0) and Robot 2 in position (-5,0)). The results can be seen in Fig. 7. It is important to notice that the robots cross it other paths but do not collide, for the instant of time is different when passing though that specific
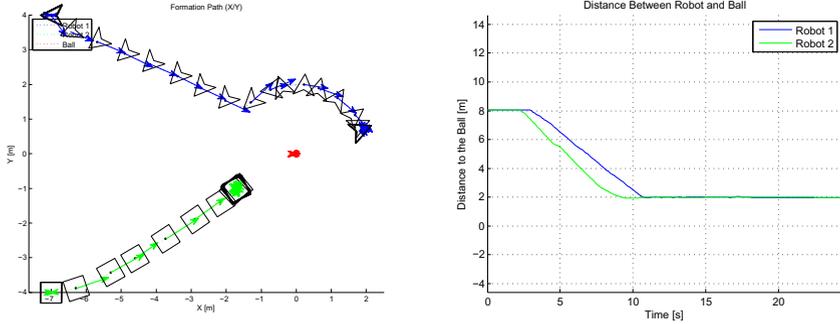
Fig. 6. Convergence into formation, simulation 2

position. The robot 2, even being closest to the ball, had to do much more turns to position itself in the desired position and orientation due to its nonholonomic constrains. While the robot number 1 goes smoothly to its position. This process takes about 22 seconds.
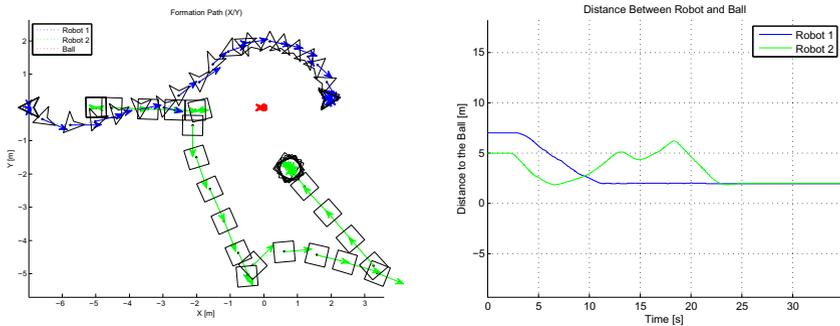

Fig. 7. Convergence into formation, simulation 3

5   Conclusions

In this paper a novel approach of a Non-linear Model Predictive Controller was presented used for multi-robot formation control. The developed framework showed to be very flexible and easily adaptable being used in holonomic and nonholonomic robots. The projected controller is capable of making a team of different robots to converge to a desired position around the target, even if the robots are very far apart. This framework could be also applied for both types of robots due to the fact that the effort of control does not take into account the tensions on the motors or the actuator input signals, but the velocities of center of mass. The results showed that the robots converged to the desired distance from the target (in this case 2m).

References

1. Chen, F., Chen, Z., Liu, Z., Xiang, L., Yuan, Z.: Decentralized formation control of mobile agents: A unified framework. Physica A: Statistical Mechanics and its Applications 387, 4917–4926 (2008)
2. Chen, Y., Wang, Z.: Formation Control: A Review and A New Consideration. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3181–3186 (August 2005)
3. Daigle, M.J., Koutsoukos, X.D., Biswas, G.: Distributed Diagnosis in Formations of Mobile Robots. IEEE Transactions on Robotics 23, 353–369 (2007)
4. Ding, Y., He, Y.: Flexible Leadership in Obstacle Environment. In: International Conference on Intelligent Control and Information Processing, pp. 788–791. IEEE Press, New York (2010)
5. Dunbar, W.B., Murray, R.M.: Model predictive control of coordinated multi-vehicle formations. In: Proceedings of the 41st IEEE Conference on Decision and Control, pp. 4631–4636. IEEE Press, New York (2002)
6. Fontes, F.A.C.C., Fontes, D.B.M.M., Caldeira, A.C.D.: Model Predictive Control of Vehicle Formations. In: Hirsch, M.J., Commander, C.W., Pardalos, P.M., Murphey, R. (eds.) Optimization & Cooperative Control Strategies. LNCIS, vol. 381, pp. 371–384. Springer, Heidelberg (2009)
7. Gu, D.: A Differential Game Approach to Formation Control. IEEE Transactions on Control Systems Technology 16, 1988–1993 (2008)
8. Kanjanawanishkul, K., Zell, A.: A model-predictive approach to formation control of omnidirectional mobile robots. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 2771–2776. IEEE Press, New York (2008)
9. Kloder, S., Hutchinson, S.: Path planning for permutation-invariant multi-robot formations. IEEE Transactions on Robotics 22, 650–665 (2006)
10. Costa, P.: Paco Wiki - SimTwo,
    `http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo`
11. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the rprop algorithm. In: IEEE International Conference on Neural Networks, pp. 586–591. IEEE Press, New York (1993)
12. Ferreira, J.R.A.: Controlo Coordenado de Equipas de Robots Móveis. Master Thesis. University of Porto, Porto, Portugal (2010)
13. Xie, F., Fierro, R.: On Motion Coordination of Multiple Vehicles with Nonholonomic Constraints. In: 2007 American Control Conference, pp. 1888–1893. IEEE Press, New York (2007)
14. Zou, Y., Pagilla, P.R., Ratliff, R.T.: Distributed Formation Control of Multiple Aircraft Using Constraint Forces. In: 2008 American Control Conference, Seattle, pp. 644–649 (2008)