



Multi-robot nonlinear model predictive formation control: Moving target and target absence



Tiago P. Nascimento^{a,*}, António Paulo Moreira^b, André G. Scolari Conceição^c

^a Department of Computer Systems, Informatics Center, Federal University of Paraíba (UFPB), Cidade Universitaria, João Pessoa, PB, Brazil

^b INESC TEC (formerly INESC Porto) and Faculty of Engineering, University of Porto, rua Dr. Roberto Frias, 4200-465 Porto, Portugal

^c LaR - Robotics Lab, Department of Electrical Engineering, Polytechnic School, Federal University of Bahia (UFBA), Rua Aristides Novis, 02 Federação, Salvador, BA, Brazil

HIGHLIGHTS

- Proposed a novel approach in formation control and active target tracking problem.
- We consider moving target and the target's presence and absence.
- The NMPFC allows the formation to converge and follow a target or a leader.
- Despite vision problems in real robots, the approach worked successfully.

ARTICLE INFO

Article history:

Received 19 November 2012

Received in revised form

2 July 2013

Accepted 12 July 2013

Available online 24 July 2013

Keywords:

Multi-robot system

Formation control

Mobile robotics

ABSTRACT

This paper describes a novel approach in formation control for mobile robots in the active target tracking problem. A nonlinear model predictive formation controller (NMPFC) for target perception was implemented to converge a group of mobile robots toward a desired target. The team must also maintain a desired formation following a target while it is moving, or follow a leader in the case of target's absence. The structure details of the controller, as well as a mathematical analysis of the formation model used, are presented. Furthermore, results of simulations and experiments with real robots are presented and discussed.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

A novel adaptive framework based in nonlinear model predictive control was conceived, in this study, and applied to the formation control of a group of mobile robots. A nonlinear model predictive control (NMPC) is used to converge a group of omnidirectional mobile robots (the 5dpo robots seen in Fig. 1) toward a desired target. The formation must be such that allows the movement of robots around the target, avoiding mates and minimizing the total amount of uncertainty in the target's perception of the group. The mobile robots team must also use the NMPC to keep adaptive formation with a moving target. Furthermore, the case of target absence where a leader robot is determined and the other robots in the group follow the leader robot using the NMPC is considered. Finally, the NMPC is implemented in a distributed fashion, embedded in each robot, and exchanging information with the

other robots in formation and it is called here nonlinear model predictive formation control (NMPFC).

This work is inserted in the active target tracking [1–4] and formation control problems [5–10]. Therefore, the problem addressed in this paper lies in the frontier between the formation control problem and the active target tracking problem. The problem here is to conceive a formation controller capable of controlling a multi-robot system in a distributive fashion considering obstacles and mates avoidance, the formation itself and the maximization of the target's observation by the group of robots in formation.

In formation control, the leader-following approach is one of the most studied [11–20]. It is based on the existence of a leader (real or virtual) that follows the precise desired trajectory while the other robots members of the formation just follow it, maintaining a preset distance and relative position. The leader robot can be a real robot with a different controller and a path generator, or it can be a virtual leader such as a target is in the target tracking problem.

One of the most used controllers in the leader–follower approach is the model predictive controller (MPC). It has been the target of studies about multi-robot motion control for almost a decade

* Corresponding author. Tel.: +55 8399115577.

E-mail address: tiagopn@ci.ufpb.br (T.P. Nascimento).



Fig. 1. The 5dpo robot.

since the first work done by [11]. After that, various works have improved the technique through time, such as [21]. Here, the authors investigated the leader-following formation control of mobile robots through the MPC. They established its control stability by adding a terminal state penalty to the cost function and a terminal state region to the optimization constraints. The authors also designed a terminal state region based on an input–output feedback linearization controller for the MPC. A suboptimal stable solution is thought to reduce the computational time used in the MPC. However, this approach has still a high computational cost.

Moreover, in 2009, [22] applied a two layer predictive controller that controls the formation of nonholonomic mobile vehicles was proposed. In their study, the authors considered that there are two sub-problems to be solved to fulfill the main goal: the trajectory control problem and the formation control problem. To solve the first sub-problem a nonlinear controller was proposed to control the robots' trajectory while a linear model predictive controller was proposed to solve the second sub-problem and control the robots' formation.

Considering the implementation of the approach using the NMPFC in a group of omnidirectional mobile robots, the next section will describe the control architecture used in this paper. In the following section, the nonlinear model predictive formation controller is presented. Furthermore, results from experiments with real robots are presented and discussed. The paper is then concluded in Section 4.

2. Nonlinear model predictive formation controller

The nonlinear model predictive control is usually implemented in a centralized fashion. It holds full knowledge of the entire system and computes all the control inputs. A centralized control using a non-convex optimization scheme applied in large-scale interconnected systems, such as water distribution systems, traffic and power systems, manufacturing systems and economic systems, may be a too complex solution or not even feasible. With the quick development of communication networks, centralized control has been gradually replaced by distributed control such as in multi-robot systems and applications in manufacturing and process industries where multiple units cooperatively produce a good. In distributed control schemes, agents share information in order to improve closed-loop performance, robustness and fault-tolerance [23]. In the approach described in this paper, the agents

(robots) share their local measurements (states) and receive the other agents' states, computing a control input using a reduced order model of the formation system dynamics. The challenge in this case is to formulate a simpler and decentralized problem which leads to a behavior that is similar to the one obtained using a centralized approach [24].

The objective of this paper is to formulate a nonlinear model predictive formation controller (NMPFC) for a multi-robot systems formation control. The general structure of a coordinated multi-robot system can be classified in three categories: distributed, centralized, or hybrid (partially distributed). These classifications are based on how the control signals of each robot are calculated. In this case, the control architecture is fully distributed [23].

The NMPFC's ability to create and maintain a formation is due to the fact that the cost functions used by the controllers of each robot in the team are coupled. The above mentioned coupling occurs when the teammates' states are used in the cost function of each robot's controller to penalize the geometry or the deviation from the desired objective. This means that the actions of each robot affect every other teammate. The optimization of a cost function that takes into account the target position and obstacle avoidance takes out the necessity of a path planner and a control loop that are usually separated in two different modules. Here there is no path planner, only the NMPFC. Each robot keeps the formation state (pose and speed of the robots in formation, and position and speed of any target that should be followed), updating them in each control loop. This information is received by the controller of each robot in the formation which in turn creates the formation geometry where the actions of each robot affect the other teammates. The NMPFC can be divided into two sub-blocks:

- *Optimizer*—This sub-block uses an online numeric minimization method to optimize the cost function and generate the signals of optimal control. The resilient propagation (RPROP) method is used here, which guaranties quick convergence [25];
- *Predictor*—The predictor performs the state evolution of the robot itself, the teammates and the target based on pre-defined models. It uses a simplified dynamic model to emulate the robot's evolution. The velocities of the teammates and target are assumed to be constant and equal to the last known velocities during the entire prediction horizon. The obstacles (moving or static) are assumed to have zero velocities during the control loop and the evolution of the relative distance between the obstacle and the robot is predicted. The predictor also emulates the evolution of the target's merged state covariance matrix.

Fig. 2 illustrates the structure of the NMPFC used in this work, where $U(k|k) = U(k) = [v_{ref}(k) \quad v_{nref}(k) \quad w_{ref}(k)]^T$ is the output control signal in the first prediction step, $\hat{U}(k+i|k)$ with $i = 0 \dots N_c - 1$ is the output control signal from the optimizer sent to the predictor, and $\hat{P}(k+i|k)$ with $i = 1 \dots N_p$ is the response of the predictor block to each $\hat{U}(k+i|k)$. Here, at an instant k , robot 1 (R_1) sends its pose $P_{R_1}(k) = [x_{R_1}(k) \quad y_{R_1}(k) \quad \theta_{R_1}(k)]^T$ to the NMPFC. Furthermore, the NMPFC also receives the other robots' poses $[P_{R_2}(k) \dots P_{R_N}(k)]$, the position of the target t in the world frame ${}^w P_t(k) = [{}^w x_t(k) \quad {}^w y_t(k)]^T$ and the velocity of the target t in the world frame ${}^w V_t(k) = [{}^w v_{x_t}(k) \quad {}^w v_{y_t}(k)]^T$.

Fig. 2 also shows the block diagram of the proposed formation control framework for robot R_1 in a formation with N robots. The subscripted R_n is used to denote the robot n where $1 \leq n \leq N$ and N is the total number of robots in formation and the subscripted t to denote the target. Each robot has an NMPFC, a cooperative target estimator (CTE) [26] and a real time data base (RTDB) communication application [27]. Other functions such as localization and the vision system are embedded in other software modules represented here as the block *Other Modules from Robot 1*. At each

and the simulation of the state evolution is given by

$$\begin{bmatrix} x_{R_n}(k) \\ y_{R_n}(k) \\ \theta_{R_n}(k) \end{bmatrix} = \begin{bmatrix} x_{R_n}(k-1) \\ y_{R_n}(k-1) \\ \theta_{R_n}(k-1) \end{bmatrix} + T \cdot \begin{bmatrix} v_{x_{R_n}}(k) \\ v_{y_{R_n}}(k) \\ w_{R_n}(k) \end{bmatrix}$$

with T as the time step and

$$\begin{bmatrix} v_{x_{R_n}}(k) \\ v_{y_{R_n}}(k) \\ w_{R_n}(k) \end{bmatrix} = \begin{bmatrix} \cos(\theta_{R_n}(k)) & -\sin(\theta_{R_n}(k)) & 0 \\ \sin(\theta_{R_n}(k)) & \cos(\theta_{R_n}(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_{\text{ref}_{R_n}}(k) \\ v_{n_{\text{ref}_{R_n}}}(k) \\ w_{\text{ref}_{R_n}}(k) \end{bmatrix}.$$

Taking into account the elements presented, the position $P_t(k)$ and velocity $V_t(k)$ of the target t (ball) in the world frame at an instant k is defined as

$$P_t(k) = [x_t(k) \quad y_t(k)]^T \quad V_t(k) = [v_{x_t}(k) \quad v_{y_t}(k)]^T \quad (5)$$

where

$$\begin{cases} x_t(k) = x_t(k-1) + T \cdot (v_{x_t}(k)) \\ y_t(k) = y_t(k-1) + T \cdot (v_{y_t}(k)) \end{cases} \quad (6)$$

and

$$\begin{cases} v_{x_t}(k) = v_{x_t}(k-1) \cdot B_{FC} \\ v_{y_t}(k) = v_{y_t}(k-1) \cdot B_{FC} \end{cases} \quad (7)$$

where B_{FC} is the ball friction coefficient.

The target's velocity unit vector is then defined as

$$\tilde{V}_t(k) = [\tilde{v}_x \ t(k) \quad \tilde{v}_y \ t(k)]^T = \frac{V_t(k)}{\|V_t(k)\|} \quad (8)$$

The position of the target relative to the robot R_n at an instant k is defined as

$$P_t^{R_n}(k) = [x_t^{R_n}(k) \quad y_t^{R_n}(k)]^T \quad (9)$$

where

$$\begin{cases} x_t^{R_n}(k) = x_t(k) - x_{R_n}(k) \\ y_t^{R_n}(k) = y_t(k) - y_{R_n}(k) \end{cases} \quad (10)$$

The unit vector which indicates the direction of the target with respect to the robot is defined as

$$\tilde{P}_t^{R_n}(k) = [\tilde{x}_t^{R_n}(k) \quad \tilde{y}_t^{R_n}(k)]^T = \frac{P_t^{R_n}(k)}{\|P_t^{R_n}(k)\|} \quad (11)$$

The bearing of the target with respect to robot R_n is defined as

$$\theta_t^{R_n}(k) = \arctan 2(y_t^{R_n}(k), x_t^{R_n}(k)). \quad (12)$$

The poses of a robot R_n relative to its teammate R_j (where $1 \leq j \leq NM$, and NM is the total number of mates) are defined as

$$P_{R_n}^{R_j}(k) = \begin{cases} x_{R_n}^{R_j}(k) = x_{R_n}(k) - x_{R_j}(k) \\ y_{R_n}^{R_j}(k) = y_{R_n}(k) - y_{R_j}(k) \end{cases} \quad (13)$$

and with respect to an obstacle O_l (where $1 \leq l \leq NO$, and NO is the total number of obstacles), it is defined as

$$P_{R_n}^{O_l}(k) = \begin{cases} x_{R_n}^{O_l}(k) = x_{R_n}(k) - x_{O_l}(k) \\ y_{R_n}^{O_l}(k) = y_{R_n}(k) - y_{O_l}(k) \end{cases} \quad (14)$$

It is important to mention that in the obstacle's state evolution, all obstacles (moving or static) are considered as having zero

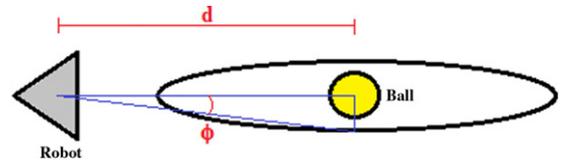


Fig. 3. Model of observation.

velocity at that time instant in order to reduce the computation load as the number of obstacles increases.

Finally, in order to model the evolution of the total amount of uncertainty with respect to the relative position between the robot and the target, a covariance model was created based on Fig. 3. This model depends on the type of camera used (such as omnidirectional mirror-camera, fish eye camera, normal direct cameras). Nevertheless, the model can easily be changed using the controller's code, allowing each robot to have a different sensor.

In the omnidirectional 5dpo robot camera, the point on the ground plane directly below the robot's catadioptric system center (also the robot's geometric center) is assumed to be the origin O of the coordinates for the model discussed in this sub-section. The target observation M_{R_n} is represented as a 2D vector $[d_t, \phi]$. d_t , where $d_t \geq 0$, $d_t \in \mathbb{R}$ is the distance to the target from O and ϕ , where $-\pi \leq \phi \leq +\pi$, $\phi \in \mathbb{R}$ is the bearing of the target from the positive X axis of the robot. The covariance model of a robot R_n in the instant k is given by (15):

$$\Sigma_{R_n}(k) = \begin{bmatrix} \sigma_{d_t}^2 & \rho \sigma_{d_t} \sigma_{\phi} \\ \rho \sigma_{\phi} \sigma_{d_t} & \sigma_{\phi}^2 \end{bmatrix} \quad (15)$$

where σ_{d_t} is the variance of the target's distance measurement d_t and σ_{ϕ} is the variance of the target's bearing measurement. ρ is the correlation coefficient. It is assumed that both the measurements are uncorrelated and $\rho = 0$.

In the case of the 5dpo robots, an empirical observation covariance model (16) was created where the variance in the target distance is directly proportional to the distance squared and the variance in the target bearing is inversely proportional to the target distance. This observation model was validated with several experiments in which it was also possible to find the values of K_a and K_b :

$$\Sigma_{R_n}(k) = \begin{bmatrix} K_a d_t^2 & 0 \\ 0 & K_b \frac{1}{d_t} \end{bmatrix}. \quad (16)$$

Furthermore, it is necessary to represent the observation covariance (16) in its canonical form in the Cartesian coordinates centered at O due to the need of ease up the covariances merging arising from the teammates. Therefore, the canonical representation of the covariance model in the direction of the target and in its perpendicular direction is given by (17):

$$\Sigma_{R_n}^{\perp}(k) = \begin{bmatrix} K_1 d_t^2 & 0 \\ 0 & K_2 d_t \end{bmatrix} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \quad (17)$$

where $K_1 = K_a$ and $K_2 = (K_b + K_a K_b)$ are constants of proportionality.

The covariance merging is performed using Smith and Cheeseman's formulation [30]. In the formation, each robot teammate covariance $\Sigma_{R_j}^{\perp}(k)$ is also predicted. The teammate's predicted covariance is also rotated in the robot's frame and then merged as per the method presented in [30]. Therefore, if it is given N robots in a formation, then R_n is the robot predicting its formation covariances and the robots R_j are the teammate with $1 \leq j \leq NM$ and $NM = N - n$ are the number of teammate in formation. The

merged covariances are given by

$$\Sigma_{\text{Merged}}^{\perp}(k) = ([\Sigma_{R_n}^{\perp}(k)]^{-1} + [\hat{\Sigma}_{R_{j=1}}^{\perp}(k)]^{-1} + \dots + [\hat{\Sigma}_{R_{j=NM}}^{\perp}(k)]^{-1})^{-1}$$

where $\hat{\Sigma}_{R_j}^{\perp}(k)$ are the covariance rotated matrices of the teammates with respect to the robot R_n .

No noise was introduced in simulation experiments. Therefore, the means of observation estimates from teammates are identical while the uncertainty ellipse around each teammate's observation is formulated as per (17) for merging.

2.2. The cost function during the presence of a target

The cost function of a NMPC (here NMPFC) represents the cost to be minimized by the predictive controller. It is typically associated with the dynamical change of the system (formation geometry) over time. Therefore, reducing the uncertainty of the target's localization and velocity estimates, while keeping the robots apart and assigning costs to the motion of the robots (for instance, to get closer to a ball). Therefore, the term (18a) of the cost function penalizes the total amount of uncertainty given by the merged covariance matrix $\Sigma_{\text{Merged}}^{\perp}(k)$. The term (18b) penalizes the distance between the target and the robot $\|P_t^{R_n}(k)\|$. To avoid collision, the penalization takes into account a threshold distance that the robot must maintain between it and the target (D_{val}).

The function (18c) penalizes the difference between the angle of the robot in the world frame (the orientation of the robot in world frame) and the angle between the robot and the target, which will allow the robot to face the ball. In the case of the 5dpo mobile robot, it faces the target when its concave face (the kicking mechanism) is toward the target. The function $\delta(\cdot)$ receives two angles as arguments and returns their difference scaled between $-\pi$ and π . The following term will influence the robot's position with respect to the target's velocity vector. It penalizes if the robot is in the wrong position during the target's movement. Here, the P_{val} will change this position and it must have a value between 1 and -1 . Note that this range of values can allow the robot to be in front of the ball, behind the ball or at its side.

The function (18e) is a potential function that penalizes the proximity between the robot and its available mates (NM). Here, the potential fields approach was used to create the mates avoidance term of the cost function. It is a modification of the previous version [31] and its weights change if the robot is too close to its mates. This is a negative linear function of distance. In the function, the given value where small distances are not penalized is $D_M = 1.5$ m. Therefore, the robots must keep a relative distance between them greater than D_M . The function (18f) works in the same way as the previous function, although it is used here to avoid obstacles. While in the last function the second sum adds until the maximum number of available mates, this function sums all obstacles in the robot's sensor range (NO). Here, $D_O = 1.5$ m has the same purpose of D_M but regarding obstacles. The NMPFC deals adequately with constraints. Although it was considered avoiding collisions using an explicit state constraint, it was not taken this choice because it would slow down the optimization algorithm.

Finally, the term (18g) penalizes the control effort. In this last function, the variation in the output control signal is penalized instead of its absolute value. Penalizing the output control signal would create a steady-state error in non-zero velocities (for example when pursuing a moving target).

The final cost function (18) is a composition of seven terms. Nevertheless, it is important to remember that here $|\cdot|$ denotes 1-norm for vector arguments and the absolute value for scalars as well as $\|\cdot\|$ represents the euclidean norm. 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 all robots, is as follows:

$$J(N_1, N_p, N_c) = \sum_{i=N_1}^{N_p} \lambda_a \times |\det(\Sigma_{\text{Merged}}^{\perp}(k+i))| \quad (18a)$$

$$+ \sum_{i=N_1}^{N_p} \lambda_0 \times |(D_{\text{val}} - \|P_t^{R_n}(k+i)\|)| \quad (18b)$$

$$+ \sum_{i=N_1}^{N_p} \lambda_1 \times |\delta(\theta_{R_n}(k), \theta_t^{R_n}(k+i))| \quad (18c)$$

$$+ \sum_{i=N_1}^{N_p} \lambda_2 \times |P_{\text{val}} + (\tilde{P}_t^{R_n}(k+i) \cdot \tilde{V}_t(k+i))| \quad (18d)$$

$$+ \sum_{i=N_1}^{N_p} \sum_{j=1}^{NM} \lambda_3 \times \max \left(\left| 1 - \frac{\|P_{R_n}^{R_j}(k+i)\|}{D_M} \right|, 0 \right) \quad (18e)$$

$$+ \sum_{i=N_1}^{N_p} \sum_{l=1}^{NO} \lambda_4 \times \max \left(\left| 1 - \frac{\|P_{R_n}^{O_l}(k+i)\|}{D_O} \right|, 0 \right) \quad (18f)$$

$$+ \sum_{i=1}^{N_c} \lambda_5 \times |\Delta U(k+i-1)| \quad (18g)$$

where N_1, N_p are the predicted horizon limits in discrete time, such that $N_1 > 0$ and $N_2 \leq 7$. $N_c = 2$ is the control horizon. $\lambda_a, \lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ and λ_5 are the weights for each component of the cost function. $\Sigma_{\text{Merged}}^{\perp}$ is the formation team's merged target observation covariance matrix. D_{val} is the threshold distance between the robot and the ball. P_{val} is the position coefficient which puts the robot around the ball in a determined position. $\Delta U(k+i-1)$ is the variation of the control signals, where $U(k)$ is the velocity vector of the robot's frame.

Several differences improve this work when compared to our previous one presented in [31]. They are as follows:

- There is a penalization function of the total amount of uncertainty of the target's perception, which allows the robots to be in the better position while converging or following a target.
- In the control effort penalization term, the 1-norm is used, which gives the controller more efficiency [32]. However, the disadvantage of 1-norm is its high nonlinearity. Given that it was adopted RPROP [25] and that it is a heuristic optimizer, it can handle the nonlinearities introduced by the use of the 1-norm.
- There was a modification on the obstacle avoidance function using a potential field approach.
- In this work, only one cost function works as a substitute for the two cost functions presented in [31], independent of the position it may assume in formation.
- Here the absence of target case was also considered.
- The results from this work proved to have an improved final convergence of the formation.

2.3. The cost function during the target's absence

The case of the target's absence is an important special case to be considered. To address this issue, another cost function similar to the previous one was created. This cost function is used only in case of the absence of a target in which each robot has to follow a robot leader which in turn performs a search in a preset area using

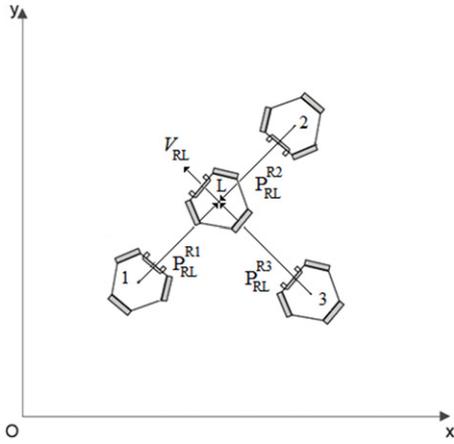


Fig. 4. Formation following a leader.

a normal reactive controller [33] with the path planner presented in [34]. Note that this cost function is used only by the follower robots.

The formation geometry, in this case, selects a fixed robot as the leader, while the other two robots become the followers. All the assumptions made for the last cost functions with respect to the target shall be made in this case with respect to the robot leader. An exception is the function (18a) due to the fact that the leader's pose is passed to the other robots (not estimated). Therefore, this cost function is a composition of six functions. The function (19a) penalizes the distance between a leader robot R_L and the follower robot $\|P_{R_n}^{R_L}(k)\|$. To avoid collision, the penalization also does not take into account a fixed distance that the robot must maintain between it and the leader robot (D_{val}).

The function (19b) penalizes the difference between the angle of the robot (a follower) in the world frame and the angle between the follower robot and the leader robot. This will make the robot face the leader while it is in movement. The last modification is in the function that will influence the robot's position with respect to the leader's velocity vector (19c) as can be seen in Fig. 4. It penalizes if the robot is in the wrong position. Here, the P_{val} will change this position and it must have a value between 1 and -1 .

The terms (19d)–(19f) of this cost function are similar to the (18e)–(18g) terms of the cost function (18). 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 both robots, is as follows:

$$J(N_1, N_p, N_c) = \sum_{i=N_1}^{N_p} \lambda_0 \times |(D_{val} - \|P_{R_n}^{R_L}(k+i)\|)| \quad (19a)$$

$$+ \sum_{i=N_1}^{N_p} \lambda_1 \times |\delta(\theta_{R_n}(k), \theta_{R_n}^{R_L}(k+i))| \quad (19b)$$

$$+ \sum_{i=N_1}^{N_p} \lambda_2 \times |P_{val} + (\tilde{P}_{R_n}^{R_L}(k+i) \cdot \tilde{V}_{R_L}(k+i))| \quad (19c)$$

$$+ \sum_{i=N_1}^{N_p} \sum_{j=1}^{NM} \lambda_3 \times \max \left(\left| 1 - \frac{\|P_{R_n}^{R_j}(k+i)\|}{D_M} \right|, 0 \right) \quad (19d)$$

$$+ \sum_{i=N_1}^{N_p} \sum_{l=1}^{NO} \lambda_4 \times \max \left(\left| 1 - \frac{\|P_{R_n}^{O_l}(k+i)\|}{D_O} \right|, 0 \right) \quad (19e)$$

$$+ \sum_{i=1}^{N_c} \lambda_5 \times |\Delta U(k+i-1)|. \quad (19f)$$

Table 1
Final weights for the cost function.

λ	Weights
λ_0	505
λ_1	918
λ_2	297
λ_3	510
λ_4	500
λ_5	5.00

3. Results

Several simulations were made to validate the NMPFC controller. Furthermore, some experiments with real robots were also made in order to see the behavior of the group under communication, vision and localization problems. Each group has convergence and retaining formation experiments. For all results the same parameters were used. Nevertheless, some assumptions must be made such as follows:

- Distance between the robot and the ball or between the robot and the leader = 1.2 m.
- Velocity of the robots in formation or following leader = 1.5 m/s in simulations and 0.7 m/s in experiments.
- Velocity of the leader = 1 m/s in simulations and 0.5 m/s in experiments.
- The RPROP parameters were the same ones used in [31] with a maximum of 20 interactions.

The final values founded through the exhaustive simulations and experiments can be seen in Table 1.

Finally, the value of the weight λ_2 was re-tuned compared to our previous work [31]. Analyzing the graphs of internal product and plot XY of the robots during the exhaustive simulations and experiments, we reached to an optimal value for this parameter. It is important to remember that this term is only active when the target (ball or robot leader) has a nonzero velocity.

3.1. Simulations

Two simulations were done for formation convergence, one for the leader following the case with the target absent and one for keeping the formation case. In the simulations, the SimTwo simulator was used [35].

3.1.1. Simulation 1: formation convergence

In the formation convergence, the target has zero velocity. Therefore, the fourth term in the cost function is equal to zero. Furthermore, without any obstacles the sixth term is also zero. The first simulation puts the robots initially as demonstrated in Fig. 5. The objective here is to make the robots converge toward the ball. The robots 1, 2 and 3, and the target's coordinates are (3.5, -1.8), (-3.5 , -1.8), (0, 3.5) and (0, 0), respectively. In Fig. 5 plot XY of the robot's movement can be seen. Note here that the robot always stops facing the ball (with the carved part toward the ball). A graph with the distance between the robot and the ball as well as the minimization of the merged covariance's determinant can be seen in Fig. 6.

3.1.2. Simulation 2: formation convergence

The second simulation places the robots initially as demonstrated in Fig. 7. The objective here is to push the robots to converge toward the ball departing from the coordinates (-5.5 , 0), (-4.5 , 0), (-3.5 , 0) and (0, 0) for the robots 1, 2, 3 and the target, respectively. This environment was created aiming to observe if the robots can avoid collisions between them. In Fig. 7, plot XY of the robot's movement can also be seen. A graph with the distance between the robot as well as the minimization of the merged covariance's determinant can be seen in Fig. 8.

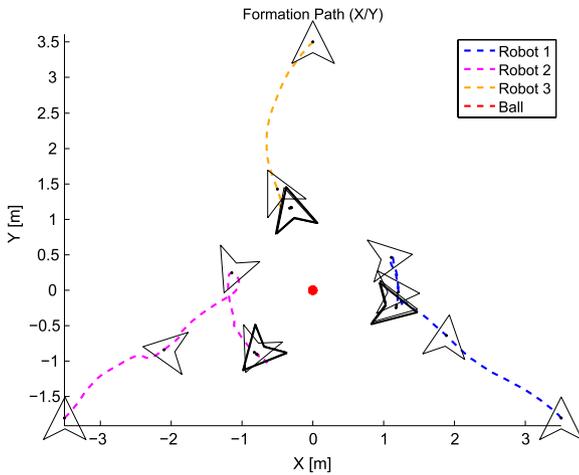


Fig. 5. Simulation 1: formation convergence—plot XY.

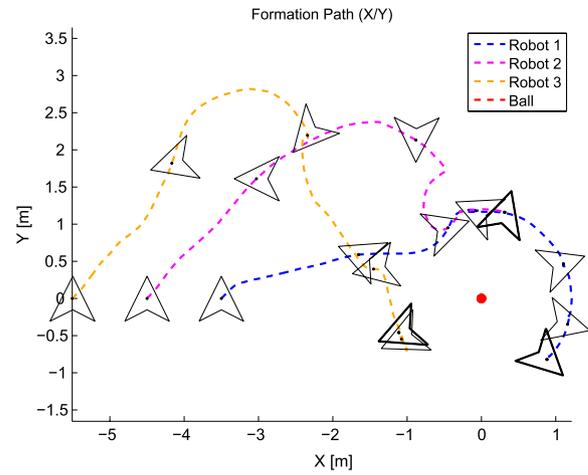


Fig. 7. Simulation 2: formation convergence—plot XY.

3.1.3. Simulation 3: the leader following

The last simulation of formation convergence positions the robots initially as demonstrated in Fig. 9. The objective here is to make the robots converge to the leader robot departing from the coordinates (4.3, -3.1), (4.3, 0), (4.3, 5.1) for the robots 2, 1 and 3, respectively, and keep a formation while following the leader robot. In this case, the target is not seen in the field, so robot 2 (as the leader robot) has an A* path planner seen in [34] with a reactive controller [33] while the follower robots (robots 1 and 3) possess the second cost function presented in this study.

In Fig. 9, plot XY of the robot’s movement and a graph with the distance between the robot and the robot leader are also presented. It also shows the graphs of the internal product between the leader and the other robots as well as the angle between them. The given P_{val} for robot 1 was $P_{val} = 1$ and for robot 3 was $P_{val} = 0$. The internal product’s graph shows here the convergence for these values. The abrupt changes in the internal product of the robots is explained by the re-orientation of the robot leader during the search. Note in Fig. 9 that the robot leader performs three big turns, or re-orientations, in its path (in coordinates (-4.3, -3), (-4.5, 3) and (4.5, 3), respectively). When the leader robot turns, the followers became oriented in 90 (°) with respect to the leader’s orientation until they converge again, also explaining the “jumps” in these graphs.

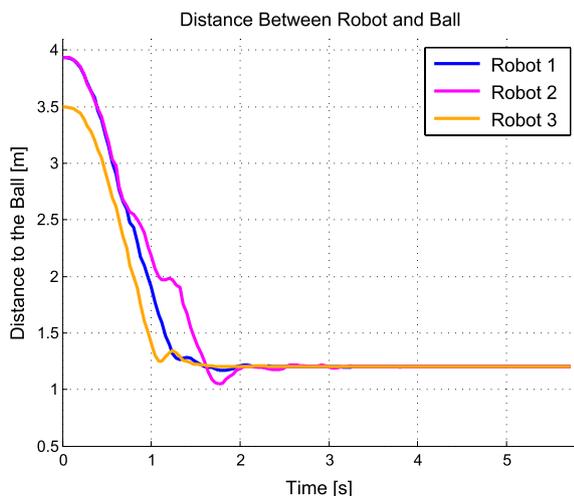


Fig. 6. Simulation 1: formation convergence—the distance between the robot and the determinant of Σ_{Merged}^{\perp} .

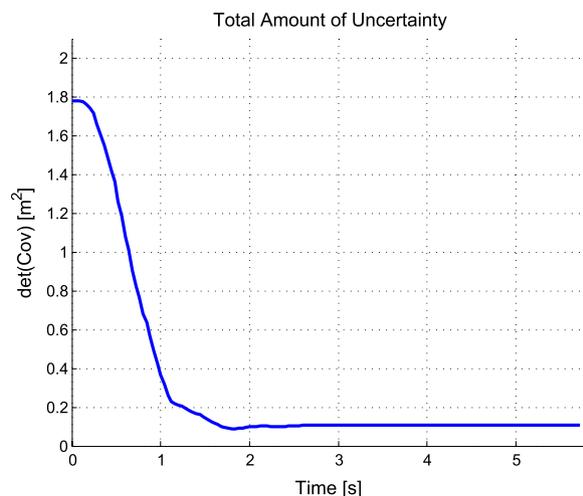
3.1.4. Simulation 4: keeping formation

In this single simulation, the fourth term in the cost function is taken into account with the movement of the target. Nevertheless, without any obstacles the sixth term is zero. This simulation places the robots initially as demonstrated in Fig. 10. The objective here is for robots to keep following the ball in a straight trajectory. Robots 1, 2, 3 and the target’s coordinates are (0, -1.5), (0, 1.5), (1.5, 0) and (0, 0), respectively. Then, a small impulse (kick) was given to the ball forcing it to possess a nonzero velocity. In Fig. 10 plot XY of the robot’s movement can be seen. A graph with the distance between the robot and the ball and the minimization of the merged covariance’s determinant can be seen in Fig. 11.

Fig. 11 also shows the graphs of the internal product between the robots and the ball as well as the angle between the robots and the ball’s velocity vector. The given P_{val} for robots 1, 2 and 3 were $P_{val} = 0$, $P_{val} = 0$ and $P_{val} = 1$, respectively.

3.2. Results of the experiments with real robots

A setup, to perform experiments with real robots, was created in order to analyze the behavior of three omnidirectional mobile robots with the NMPFC. An external computer connected to a router by a cable was needed in order to serve as a bridge of shared information through RTDB to all the robots, where the log could be collected from the robots. Therefore, to run the experiments



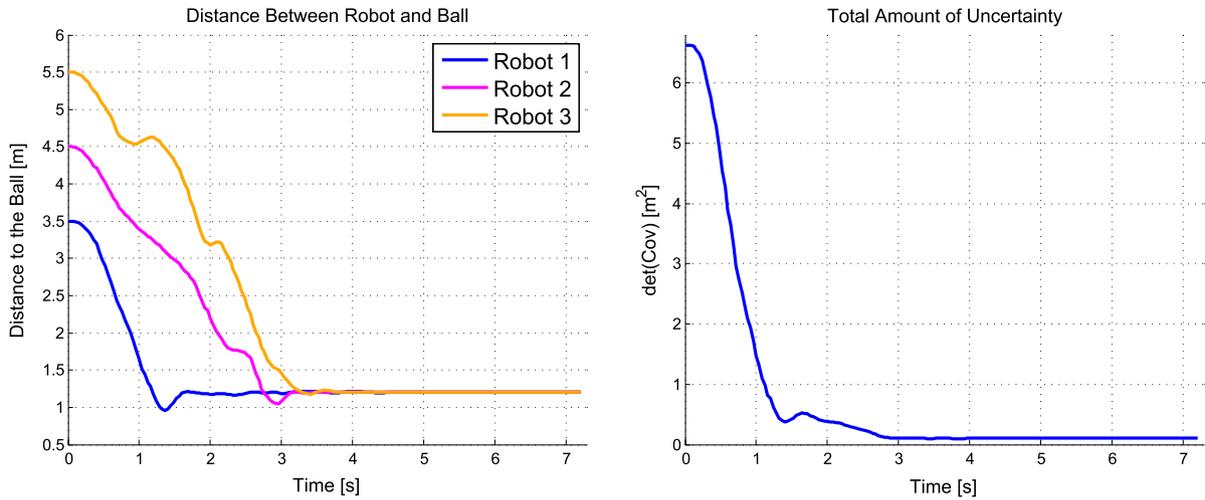


Fig. 8. Simulation 2: formation convergence—the distance between the robot and the determinant of $\Sigma_{\text{Merged}}^{\perp}$.

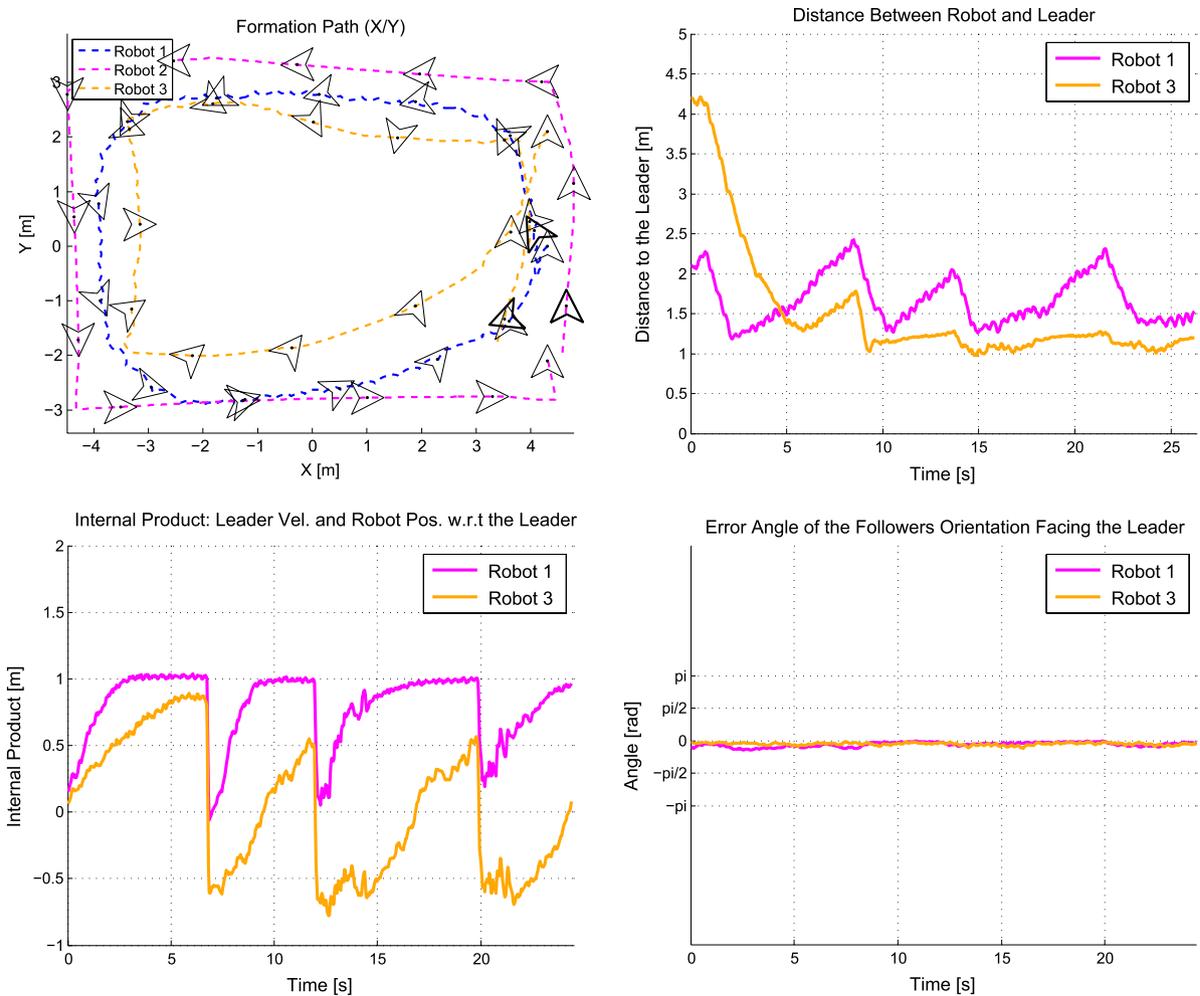


Fig. 9. Simulation 3: the leader following—plot XY and distance between the leader robot and the followers (top), the internal product between leader and followers and the error angle of the followers orientation facing the leader (bottom).

a workstation (Intel Core i7 3 GHz/Core with 8 Gb RAM) with Ubuntu 9.04 that runs said bridge was used. Each robot had a computer, a Notebook (Intel Dual Core 2 GHz/Core with 2 Gb RAM) with Ubuntu 9.04, running its own NMPFC, CTE and RTDB applications previously seen in Fig. 2. Finally, the experiments with

real robots were executed in order to repeat the environment created in the simulations with two experiments in formation convergence, one for the leader following the case with the target absent and one for keeping the formation case. Videos from all experiments are attached to this paper which is available in

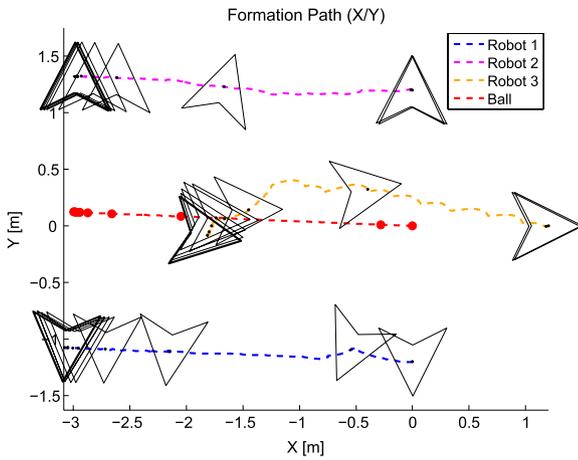


Fig. 10. Simulation 4: keeping formation—plot XY.

this experiment and the objective is to see the convergence toward a target placed in the center of the field. Robots 1, 2 and 3 are placed far from each other and their positions are $(-3, 1.7)$, $(0, -2)$ and $(3, 1.7)$, respectively. All robots were 270° oriented in the world frame. The target's coordinates are $(0, 0)$. In Fig. 13 the graph of the distance between the robot and the ball as well as the graph of the total amount of uncertainty (merged covariances) minimization can be seen. Video 1 shows robots' movement as presented in plot XY.

Note here that no collisions between the robots occur and that the robot always stops while facing the ball (with the carved part toward the ball) as shown in the simulations. As it can be seen, the ball “jumps” from one coordinate to another because the formation chooses the “best” ball amongst the balls' measurement which come from all three robots. Nevertheless, the robots move in a perfect circle around the ball while trying to minimize the covariance. That is explained by the fact that even if the best ball changes, or “jumps” from one coordinate to another, the total cost of these terms (covariance and distance) are kept very similar as it can be seen in Fig. 13. However, the formation converges successfully.

3.2.2. Real experiment 2: formation convergence

The second experiment places the robots initially at the coordinates $(2, 0)$, $(3, 0)$, $(4, 0)$ and $(0, 0)$ for robots 1, 2, 3 and the target respectively, as demonstrated in Fig. 14. The objective here is to observe the robots converging to the ball avoiding collisions between them. Similarly to the second pair of simulations, the robots

<http://dx.doi.org/10.1016/j.robot.2013.07.005> to aid in the analysis of the robots' behavior during the tests.

3.2.1. Real experiment 1: formation convergence

The real experiments were conducted with robots number 1, 2 and 3. The first experiment placed the robots initially as demonstrated in Fig. 12. In this figure, plot XY of the robot's movement can also be seen. As it was in the first simulation, there is no obstacles in

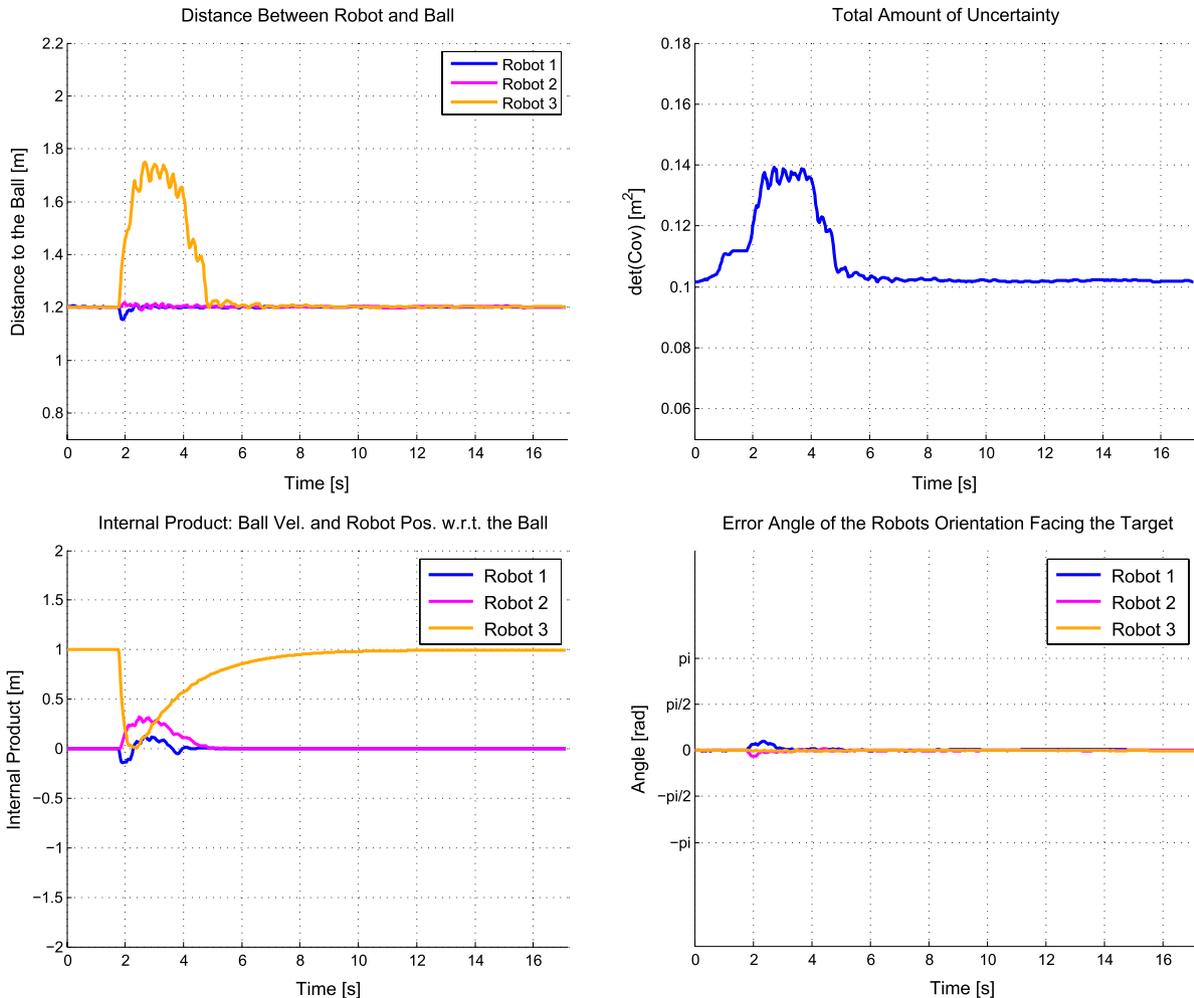


Fig. 11. Simulation 4: keeping formation—distance between the robot and the ball and the determinant of Σ_{Merged}^\perp (top), the internal product between the robots and the ball and the error angle of the robots orientation facing the target (bottom).

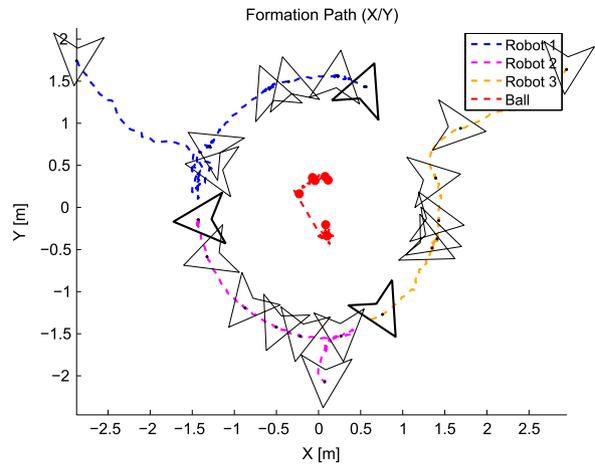


Fig. 12. Real experiment 1: formation convergence.

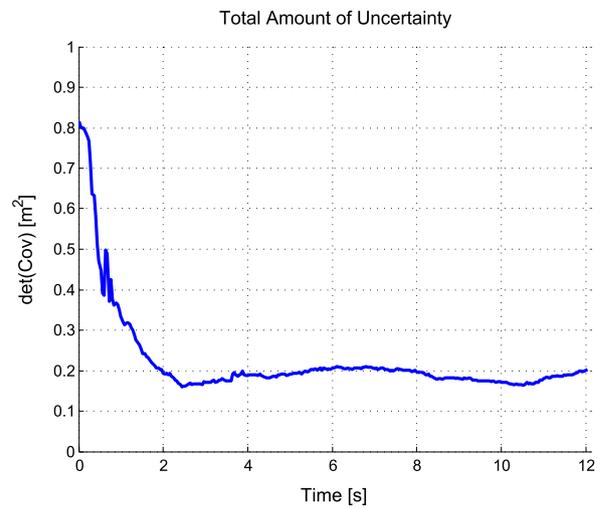
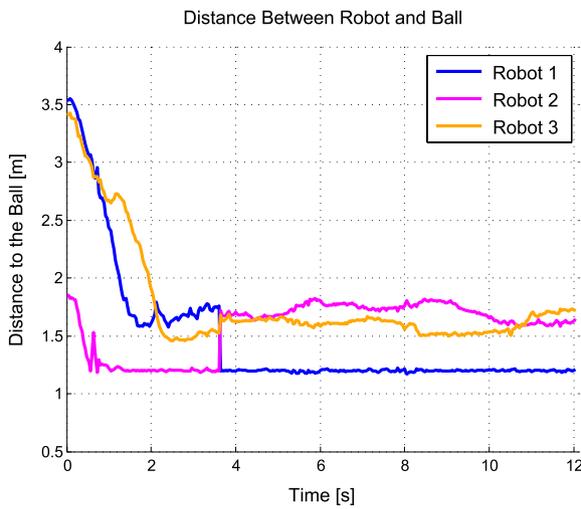


Fig. 13. Real experiment 1: formation convergence—the distance between the robot and the ball and the determinant of $\Sigma_{\text{Merged}}^{\perp}$.

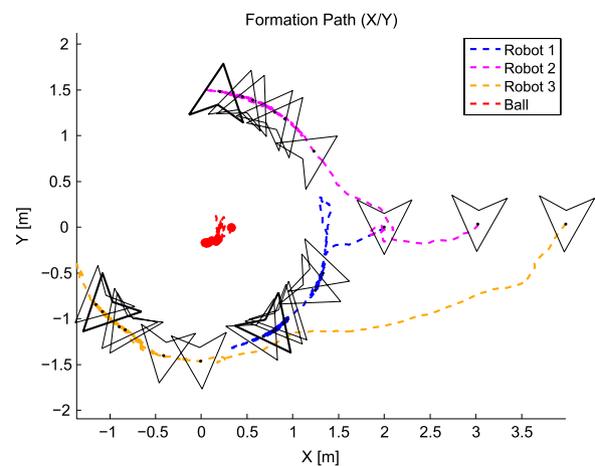


Fig. 14. Real experiment 2: formation convergence.

are placed in a horizontal line to increase the difficulty in the convergence forcing the robots to avoid each other. Video 2 shows the robots' movement successfully converging toward the target while avoiding collision between them.

In Fig. 15, a graph with the distance between the robot and the ball as well as the graph of the total amount of uncertainty (merged

covariances) minimization can also be seen. Note that this time, there is no “jump” on the “best” ball.

3.2.3. Real experiment 3: the leader following

This experiment, like the leader following simulation, placed the robots initially as demonstrated in Fig. 16. The objective here

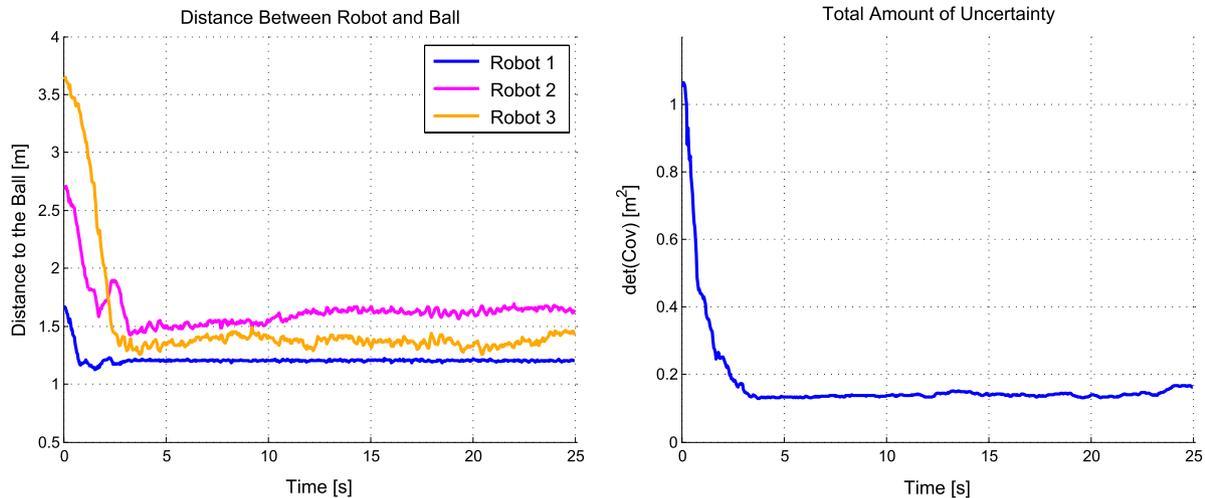


Fig. 15. Real experiment 2: formation convergence—the distance between the robot and the ball and the determinant of $\Sigma_{\text{Merged}}^{\perp}$.



Fig. 16. Real experiment 3: the leader following.

is to converge the two follower robots toward the leader robot departing from the coordinates (3, -2), (3, 0), (3, 2) for robots 1, 2 and 3, respectively. The follower robots must also keep a formation while following the leader.

Therefore, robot 1 (as the leader robot) has an A* path planner with a reactive controller while the follower robots (robots 2 and 3) have the second cost function presented in this paper. In Fig. 17 plot XY of the robot's movement as well as a graph with the distance

between the follower robots and the robot leader can be seen. Video 3 shows the robots' movement as presented in plot XY.

Fig. 18 shows the graphs of the internal product between the leader and the other robots as well as the angle between them. Finally, the given value for P_{val} for robots 2 and 3 were $P_{\text{val}} = -1$ and $P_{\text{val}} = 0$, respectively. Similarly to the simulations and despite the correct orientation toward the leader robot, the followers have trouble in converging the internal product due to the reasons previously explained. The abrupt changes in the robots' internal product are also explained by the same reasons demonstrated in the simulation of this experiment.

3.2.4. Real experiment 4: keeping the formation

A last experiment was executed to perform formation maintenance with real robots. As the ball has a nonzero velocity, the fourth term in the cost function is taken into account. The real experiment consisted of pushing the ball in a straight line with the help of a stick to emulate a small kick. The robots 1, 2, 3 and the target's initial coordinates are (0, 1.5), (0, -1.5), (-1.5, 0) and (0, 0), respectively, as demonstrated in plot XY of the robot's movement in Fig. 19. Video 4 shows the robots' movement as presented in plot XY.

In the formation, each robot has, with a small amount of errors in the observation, its own measurement of the target in the world's frame. During the formation, this information is exchanged and a "best ball" is chosen from all the measurements that come

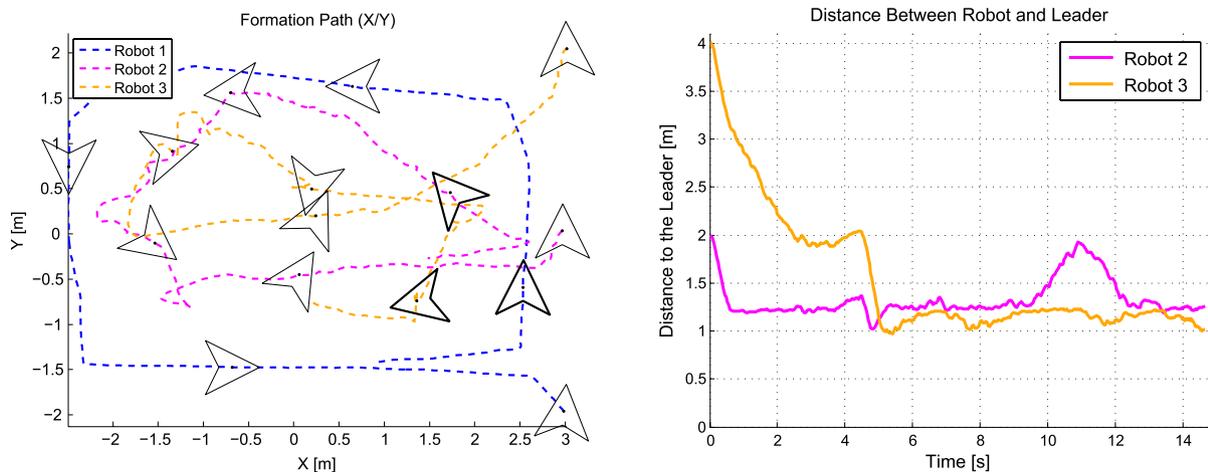


Fig. 17. Real experiment 3: the leader following—plot XY and the distance between the leader robot and the followers.

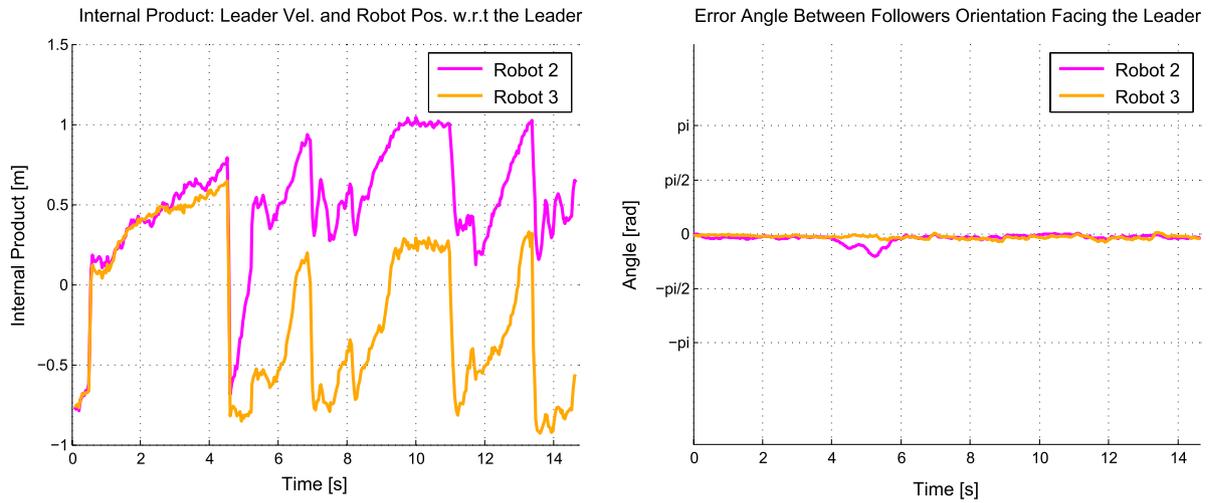


Fig. 18. Real experiment 3: the leader following—the internal product between the leader and followers and the error angle of the followers orientation facing the leader.

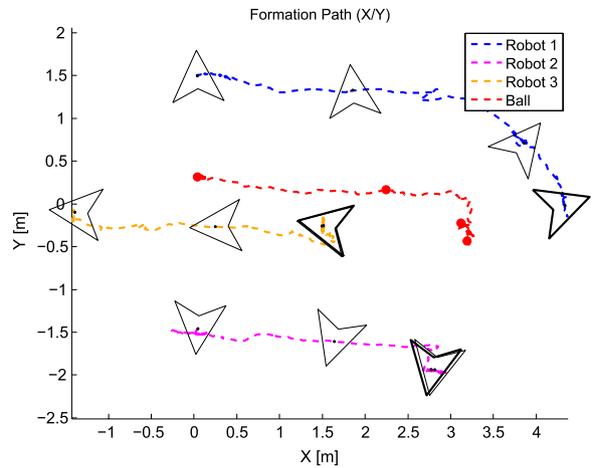


Fig. 19. Real experiment 4: keeping the formation.

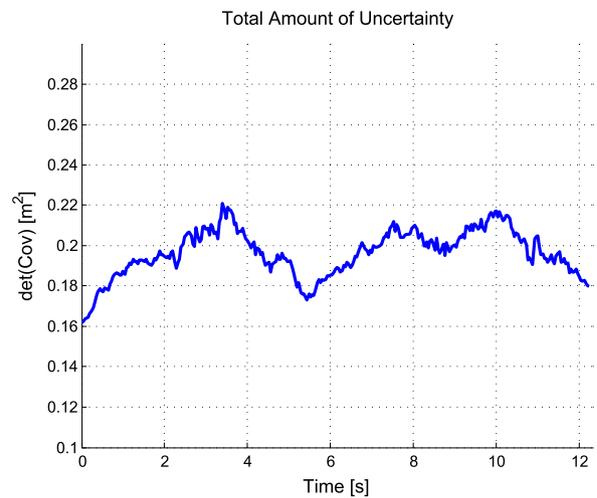
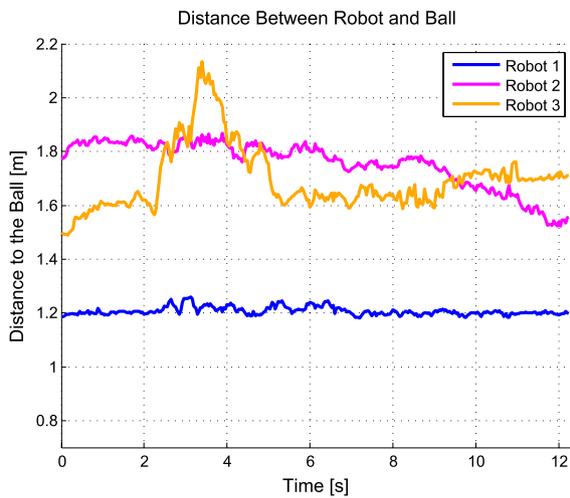


Fig. 20. Real experiment 4: keeping the formation—plot XY and the determinant of Σ_{Merged}^{-1} .

from the robots. As a “best ball” is chosen between the measured balls, the position of this final ball can create, along with small errors in the localization, differences in the distances between the robots and the ball as it can be seen in plot XY and in Fig. 20 that shows the distance between the robots and the ball and the graph

of the total amount of uncertainty (merged covariances) minimization.

Finally, in Fig. 21 the graphs of the internal product between the robots and the ball as well as the angle between the robots and the ball’s velocity vector can be seen. The same values of P_{val} for robots

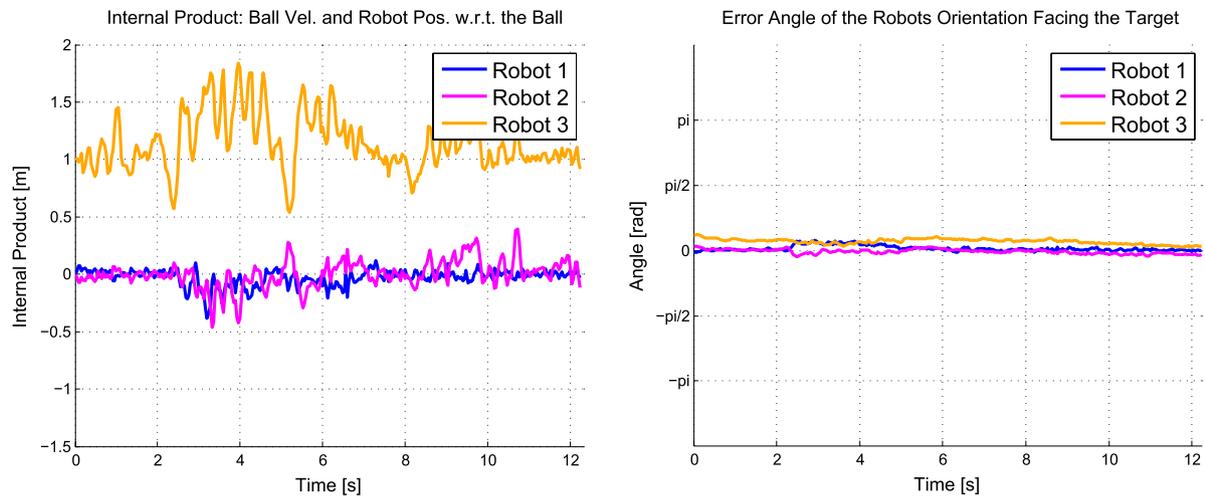


Fig. 21. Real experiment 4: keeping the formation—the internal product between the robots and the ball and the error angle of the robots orientation facing the target.

1, 2 and 3 used in the corresponding simulation were used in this experiment.

4. Conclusion

This paper had the main objective to present a new nonlinear model predictive controller used in formation control of multi-robot systems, here called a nonlinear model predictive formation controller (NMPFC). The NMPFC was implemented in order to converge a group of mobile robots toward a desired target and also to maintain the formation during the target's movement. Furthermore, a methodology to empirically find the initial values for the weights of the NMPFC cost function was presented. Finally, the case of the absence of a target was considered too, choosing a fixed leader robot in this case and using a second cost function to follow the leader robot. The details of the controller structure as well as a mathematical analysis of the formation model were presented.

This paper presented several novel contributions introduced by this work in formation control using a nonlinear model predictive controller, such as the distributive architecture to control robots in formation; a generic cost function for penalization of undesirable behaviors of the formation when following an observed and non-controlled target; the minimization of a merged covariances of the target perception in the cost function; and the controller's robustness despite vision and localization problems present in real robots environments.

The results also showed the influence of the vision system in the controllers efficiency during the formation keeping experiments. Small errors in the ball detection and localization are reflected in the calculation of the distance between the robot and the target and in the covariance minimization. Nevertheless, the controller was successful in converging the robots to the desired pose, minimizing the total amount of uncertainty while avoiding mates and setting the correct pose for all robots. Both convergence and keeping the formation were such that the total amount of uncertainty in the target's perception was minimized. Regarding the experiments without a target, the follower robots were successful in converging toward the leader and keeping the formation while the leader robot was moving.

It can be concluded that the NMPFC controller demonstrated a good performance in formation control of multi-robot systems. It could be implemented in different types of robots and penalize several non-desirable behavior of the formation with the same generic structure. Nonlinearities could be included in the prediction model such as the total amount of uncertainty in the target's perception which in turn gave relevance to the requirements imposed

by target localization and/or tracking. Furthermore, energy costs, avoidance of obstacles, uncertainties and formation characteristics were considered in a single optimization problem. Nevertheless, among the disadvantages, the NMPFC still possesses reasonable computational costs in the system's processing. Finally, there is no guarantee of global minimization of the cost function during the optimization process. The stability analysis was not considered in this study as well. Nevertheless, the system is BIBO stable, for the robot's velocity was limited on 0.7 m/s where it could achieve up to 2 m/s for safety reasons.

References

- [1] F. Morbidi, G.L. Mariottini, On active target tracking and cooperative localization for multiple aerial vehicles, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, USA, 2011, pp. 2229–2234.
- [2] F. Morbidi, C. Ray, G.L. Mariottini, Cooperative active target tracking for heterogeneous robots with application to gait monitorin, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, USA, 2011, pp. 3608–3613.
- [3] K.X. Zhou, S.I. Roumeliotis, Multi-robot active target tracking with distance and bearing observations, in: The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, Saint Louis, USA, 2009, pp. 2209–2216.
- [4] K. Zhou, S.I. Roumeliotis, Multirobot active target tracking with combinations of relative observations, *IEEE Transactions on Robotics* 27 (4) (2011) 678–695.
- [5] R.M. Murray, Recent research in cooperative control of multivehicle systems, *Journal of Dynamic Systems, Measurement, and Control* 129 (3) (2007) 571–583.
- [6] S. Monteiro, E. Bicho, Robot formations: robots allocation and leader–follower pairs, in: Proceedings of the 2008 IEEE International Conference on Robotics and Automation, ICRA2008, Pasadena, CA, USA, 2008, pp. 3769–3775.
- [7] H. Lim, Y. Kang, J. Kim, C. Kim, C. Kim, Multirobot active target following unmanned ground vehicles using nonlinear model predictive control, in: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Singapore, 2009, pp. 945–950.
- [8] S. Monteiro, E. Bicho, Attractor dynamics approach to robot formations: theory and implementation, *Autonomous Robots* 29 (3) (2010) 331–355.
- [9] D.H.A. Maithripala, D.H.S. Maithripala, S. Jayasuriya, A geometric approach to dynamically feasible, real-time formation control, *Journal of Dynamic Systems, Measurement, and Control* 133 (2) (2011) 1–14.
- [10] H. Yang, F. Zhang, Robust control of formation dynamics for autonomous underwater vehicles in horizontal plane, *Journal of Dynamic Systems, Measurement, and Control* 134 (2) (2012) 1–7.
- [11] W.B. Dunbar, R.M. Murray, Model predictive control of coordinated multi-vehicle formations, in: Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, Nevada, USA, 2002, pp. 4631–4636.
- [12] Y. Li, X. Chen, Leader-formation navigation with sensor constraints, in: Proceedings of the 2005 IEEE International Conference on Information Acquisition, Hong Kong and Macau, China, 2005, pp. 554–559.
- [13] H. Fukushima, K. Kon, F. Matsuno, Distributed model predictive control for multi-vehicle formation with collision avoidance constraints, in: Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 2005, pp. 5480–5485.
- [14] Y. Hao, S.K. Agrawal, Planning and control of UGV formations in a dynamic environment: a practical framework with experiments, *Robotics and Autonomous Systems* 51 (1) (2005) 101–110.

- [15] Y. Gu, B. Seanor, G. Campa, M.R. Napolitano, L. Rowe, S. Gururajan, S. Wan, Design and flight testing evaluation of formation control laws, *IEEE Transactions on Control Systems Technology* 108 (6) (2006) 1105–1112.
- [16] J. Shao, G. Xie, L. Wang, Leader-following formation control of multiple mobile vehicles, *IET Control Theory & Applications* 1 (2) (2007) 545–552.
- [17] L. Shi-cai, T. Da-long, L. Guang-Jun, Formation control of mobile robots with active obstacle avoidance, *Automatica* 33 (5) (2007) 529–535.
- [18] B. Liu, T. Chu, L. Wang, G. Xie, Controllability of a leader–follower dynamic network with switching topology, *IEEE Transactions on Automatic Control* 53 (4) (2008) 1009–1013.
- [19] L. Consolini, F. Morbidi, D. Prattichizzo, M. Tosques, Stabilization of a hierarchical formation of unicycle robots with velocity and curvature constraints, *IEEE Transactions on Robotics* 25 (5) (2009) 1176–1184.
- [20] W. Ni, D. Cheng, Leader-following consensus of multi-agent systems under fixed and switching topologies, *Systems & Control Letters* 59 (3–4) (2010) 209–217.
- [21] D. Gu, E. Yang, A suboptimal model predictive formation control, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, USA, 2005, pp. 3646–3651.
- [22] D.M. Fontes, F.A. Fontes, A. Caldeira, Model predictive control of vehicle formations, in: M.J. Hirsch, C.W. Commander, P. Pardalos, R. Murphey (Eds.), *Optimization and Cooperative Control Strategies*, in: *Lecture Notes in Control and Information Sciences*, Springer Verlag, 2009, pp. 371–384.
- [23] J.M. Maestre, D.M. de la Pena, E.F. Camacho, Distributed model predictive control based on a cooperative game, *Optimal Control Applications and Methods* 32 (2) (2011) 153–176.
- [24] K. Kanjanawanishkul, Coordinated path following control and formation control of mobile robots, Ph.D., Eberhard-Karls-Universität Tübingen, 2010.
- [25] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the rprop algorithm, in: *IEEE International Conference on Neural Networks*, San Francisco, CA, USA, 1993, pp. 586–591.
- [26] A. Ahmad, P. Lima, Multi-robot cooperative object tracking based on particle filters, in: *5th European Conference on Mobile Robots*, Örebro, Sweden, 2011, pp. 1–6.
- [27] L. Oliveira, L. Almeida, F. Santos, A loose synchronisation protocol for managing RF ranging in mobile ad-hoc networks, in: *RoboCup 2011: Robot Soccer World Cup XV*, in: *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, Istanbul, Turkey, 2012, pp. 574–585.
- [28] T.P. Nascimento, P. Costa, P. Costa, A.G.S. Conceição, A.P.G.M. Moreira, Modeling omnidirectional mobile robots: an approach using SimTwo, in: *Proceedings of the 10th Portuguese Conference on Automatic Control*, Funchal, Portugal, 2012, pp. 117–122.
- [29] A.G.S. Conceição, A.P.G.M. Moreira, P. Costa, A nonlinear mobile robot modeling applied to a model predictive controller, in: *Proceedings of the 2009 ACM symposium on Applied Computing*, Honolulu, Hawaii, USA, 2009, pp. 1186–1187.
- [30] R.C. Smith, P. Cheeseman, On the representation and estimation of spatial uncertainty, *The International Journal of Robotics Research* 5 (4) (1986) 56–68.
- [31] T.P. Nascimento, A.G.S. Conceição, F.A. Fontes, A.P.G.M. Moreira, Leader following formation control for omnidirectional mobile robots: the target chasing problem, in: *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics*, ICINCO, Noordwijkerhout, Netherlands, 2011, pp. 102–111.
- [32] E.F. Camacho, C. Bordons, *Model Predictive Control*, Springer, London, England, 2004.
- [33] T.P. Nascimento, A.G.S. Conceição, A.P.G.M. Moreira, Omnidirectional mobile robot's multivariable trajectory tracking control: a robustness analysis, in: *9th Portuguese Conference on Automatic Control*, Coimbra, Portugal, 2010, pp. 63–68.
- [34] T.P. Nascimento, A.G.S. Conceição, A.P.G.M. Moreira, A modified A* application to a highly dynamic unstructured environment, in: *Proceedings of the 11th International Conference on Mobile Robots and Competitions*, Lisbon, Portugal, 2011, pp. 9–14.
- [35] P.J. Costa, Simtwo, 2010. Available from <http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo> URL: <http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo>.



IEEE RAS since 2004.



Portuguesa de Robótica" and member of IEEE and IEEE RAS.



André G. Scolari Conceição received the B.S. and M.S. degrees in Electrical Engineering from Pontifical Catholic University of Rio Grande do Sul (Brazil) in 2001 and 2004, respectively, and the Ph.D. degree from Oporto University (Portugal) in Electrical and Computer Engineering, in 2007. In 2009, he joined the Department of Electrical Engineering, Federal University of Bahia, Salvador (Brazil), as an Assistant Professor. His main research interests are in mobile robotics, multi-robots systems, sensor fusion, state and parameter estimation, process control, and vehicle dynamics.