Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A decentralized Polynomial based SLAM algorithm for a team of mobile robots

Luigi D'Alfonso, Antonio Grano*, Pietro Muraca, Paolo Pugliese

DIMES, Università della Calabria, 87036 Rende (Cs), Italy,
{ldalfonso, agrano, muraca, pugliese}@dimes.unical.it.
**A. Grano is supported by RISPEISE-POR Calabria FSE 2007/2013*

Abstract: In this paper a novel solution to the Simultaneous Localization and Mapping (SLAM) problem for a team of mobile robots is proposed. The algorithm aims at approximating the robots surrounding environment by a set of polynomials, ensuring high mapping performance and low communication cost. To this sake, once two robots meet, only the acquired polynomials data are exchanged. Also, the algorithm has been developed trying to minimize each robot computational requirements so that it can be implemented in a decentralized way. Numerical simulations are reported to show the effectiveness of the proposed solution.

Keywords: SLAM; Localization; Mapping; Sensor fusion; Extended Kalman Filter.

1. INTRODUCTION

The problem of simultaneously localizing a mobile robot and building a map of its surrounding environment has been introduced for the first time by Smith et al (1986) and since then it has received considerably attention. Until recently, most research on this topic has involved a single mobile robot; on the other hand, using a team of robots to solve the SLAM problem allows to achieve the mapping goal faster and, hopefully, with a better performance w.r.t. the single robot case (Fox et al (2006)).

The main advantage of the multi-robot SLAM is the data exchange between robots. When two robots meet, they can exchange the currently acquired maps: if the two robots have explored and mapped different parts of the environment, after the data exchange each of them also knows the parts detected by the other. However, a team of coordinated mobile robots also introduces several sources of complexity: limited bandwidth; unreliable wireless communication channels; team coordination managing; shared map managing between robots; memory requirements (depending on the number of robots and the map size).

Multi-robot SLAM algorithms can be divided into two main groups: centralized and decentralized ones. In the first group, the main computations are performed by a central unit receiving the information acquired by the robots. The second group is related to decentralized algorithms where each robot makes its own computations and shares part of its own information only with the nearest robots. A centralized approach usually ensures better results, but if the central unit breaks, the whole algorithm fails. On the contrary, in a decentralized context, the algorithm works also if one of the teammates has a failure.

Thrun (2001) proposes a decentralized SLAM algorithm combining fast maximum likelihood map growing with a Monte Carlo localizer based on particle representation. Authors make the restrictive assumptions of known relative robots initial positions and of overlap in robots maps. Howard et al (2006) propose a centralized structure for a team of mobile robots, which solves the mapping problem using a manifold based representation for two dimensional maps. The advantage is self-consistency when closing loops: maps are not affected by cross over problem.

Zhou et al (2006) present a distributed Extended Kalman Filter (EKF) algorithm to build the local feature map for each teammate. The local maps are merged into a single global map after rendezvous events between robots. Tong et al (2008) propose a centralized multi-robot SLAM solution based on an EKF that estimates a state vector collecting the poses of the robots and the locations of the observed landmarks. Carbone et al (2011) investigate the SLAM problem for a multi-robot system relaxing the assumptions made in Thrun (2001) and proposing an application of Rao-Blackwellized Particle Filters for the purpose of cooperatively estimating SLAM posterior; each robot travels independently and each pair of robots exchange the acquired information once they meet.

All the cited solutions suffer of high costs in terms of energy consumption, since a large amount of information must be transferred between the robots (decentralized approaches) or between the robots and the central unit (centralized approaches). Data transferring is the most energy consuming operation in a multi-robot SLAM algorithm, thus communications have to be correctly managed, and the amount of exchanged data must be kept to a minimum.

The goal of this paper is to solve the SLAM problem using a team of mobile robots with little data communications. We propose a decentralized solution based on approximating the environment boundaries by a set of polynomials. By the simple mathematical characterization of a polynomial, exchanging such data between two robots requires a low communication cost, saving robots' batteries.

The paper is organized as follows: in Section 2 the problem statement is given; in Section 3 the multi-robot SLAM

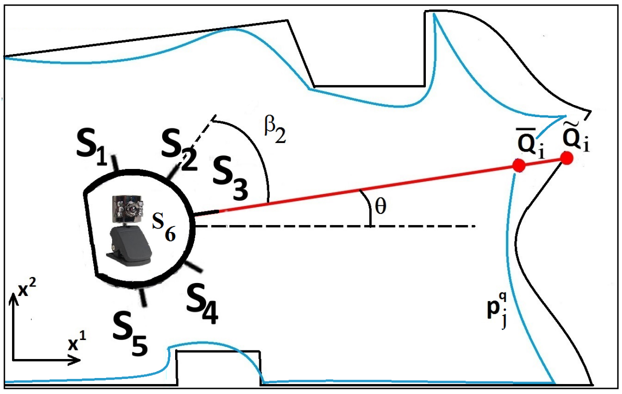

Fig. 1. Real environment (black line), polynomial based modeled environment (light blue line).

algorithm is described; in Section 4 numerical simulations are shown and finally in Section 5 some conclusions are drawn.

2. PROBLEM STATEMENT

Assume a set of N mobile robots placed in an unknown environment is given, each robot equipped with n_S sensors S_i, $\{i = 1, \dots, n_S\}$, able to provide its distance from the environment boundaries and from the other robots [1]. Consider one of the mobile robots, r_j, involved into the SLAM algorithm; Fig. 1 shows the robot for the case $n_S = 6$ (five distance sensors and one camera). Let $(x^1_{j,k}, x^2_{j,k})$ be the robot center coordinates at step k and $\theta_{j,k}$ be the robot heading w.r.t. the robot axis (orthogonal to the wheels axis). Whatever is the robot type, it can be modeled through a set of non linear difference equations

$$x_{j,k+1} = f(x_{j,k}, u_{j,k}) + w_{j,k} \tag{1}$$

where $x_{j,k} = [x^1_{j,k}, x^2_{j,k}, \theta_{j,k}]^T$ is the state of r_j at time k, the robot input is $u_{j,k}$ and $w_{j,k} = [w^1_{j,k}, w^2_{j,k}, w^\theta_{j,k}]^T$ is a Gaussian noise with zero mean and covariance matrix W_j, which also takes into account for unmodeled dynamics.

Since there is no *a-priori* knowledge about the environment, a model for its boundaries is required. Following D'Alfonso, Grano et al (2013), boundaries will be modeled by a set of m-th order polynomials, as shown in Fig. 1. At step k, each robot r_j collects the polynomials that currently map its surrounding environment in the array $\mathbb{E}_{j,k} = \{p^q_{j,k}\}_{q=1}^{n_{j,k}}$: see Fig. 1; in the following, to keep the notation clear, where possible some indexes will be omitted, implicitly relating all the variables to robot r_j. Neglecting the time dependency, each m-th order polynomial $p(\xi) = \sum_{i=1}^{m} b^q_i \xi^i + c^q_0$ is represented by a coefficient c^q_0, related to the polynomial position, and by a set of coefficients $\{b^q_m, b^q_{m-1}, \cdots, b^q_1\}$ related to the polynomial shape.

As shown in Fig. 1 for sensor S_3, the measurement taken by sensor S_i is the distance from the robot center to one point on the environment boundaries, $\tilde{Q}_i = (\tilde{x}^1_i, \tilde{x}^2_i)$. This measurement is modeled through the distance from the robot center to the intersection point, $\overline{Q}_i = (\overline{x}^1_i, \overline{x}^2_i)$, between the axis of sensor S_i and one of the environment modeling polynomials. As shown by D'Alfonso, Grano et al (2013), since the orientation β_i of each sensor w.r.t. robot axis is known, measurements are a function of the robot state and of the polynomials intercepted by sensors beam:

$$y_k = h(x_{j,k}, \overline{p}_k) + v_{j,k}, \tag{2}$$

where the size of vector y_k is n_S, $\overline{p}_k$ collects the polynomials intercepted by each sensor axis at time k and $v_{j,k}$ is the measurement noise, assumed to be a zero mean Gaussian noise with covariance V_j and uncorrelated with $w_{j,k}$. The noises, $v_{j,k}$ and $w_{j,k}$, affecting each robot r_j are assumed to be uncorrelated with the noises affecting the other robots.

Goal of this work is to estimate the position and orientation of each robot, $x_{j,k} = [x^1_{j,k}\ x^2_{j,k}\ \theta_{j,k}]^T, j = 1, \dots, N$, and to simultaneously find the best polynomial approximation of each of the environment portions detected by the robots, i.e., $\mathbb{E}_{j,k}, j = 1, \dots, N$.

3. MULTI-ROBOT SLAM

Two main parts may be detected into a multi-robot SLAM algorithm: (1) the part regarding the single robot behavior and (2) the part about the robots behavior when they meet. Regarding the first part, since the robot is equipped with its own sensors, it is able to build a map and localize itself on its own. As for the second part, the idea is to use the data exchanged between two robots once they meet and are involved into a *rendezvous* event.

Each SLAM algorithm is based on an estimation algorithm to localize the robot and simultaneously build the environment map. The most used SLAM estimator in the literature is the Extended Kalman Filter (EKF), which can be easily adapted to the SLAM context by defining an augmented state containing both the robot pose and the environment landmarks.

3.1 Single robot SLAM

In the following, for the sake of simplify notation, the subscript index j will be omitted, assuming all the variables implicitly related to the robot r_j. The proposed single robot SLAM approach is based on the use of two types of landmarks: shape landmarks and innovation landmarks. The first ones are the modeling polynomials coefficients; as proposed by D'Alfonso, Grano et al (2013), they are used as landmarks into the SLAM landmark extraction and data association steps. Innovation landmarks are the polynomials position coefficients, and they are inserted into the SLAM augmented state and used into the Kalman filter steps. The resulting augmented state is, for each robot, $X_k = [x^T_k, c^1_{0,k}, \dots, c^{n_k}_{0,k}]^T$.

This strategy comes spontaneously as we use distance sensors: the Kalman Filter innovation term is used to change the polynomials position coefficients, moving each mapping polynomial towards or away from the robot, while keeping its shape. In other words, the landmark extraction process is used to get the shape of the polynomials while their positions are changed according to the Kalman Filter.

Following the same lines in D'Alfonso, Grano et al (2013), the resulting EKF based SLAM algorithm contains a SLAM step made of a landmark extraction and data association task and an update task. The goal of the proposed landmark extraction and data association algorithm is to

[1] For example, cameras could be used and each robot could be marked with a different marker.

obtain an accurate environment approximation by storing the smallest possible amount of data and by requiring a small enough computational cost to elaborate them during robot motion. In the following, for the sake of simplicity, the computations will be referred to x^1-variate polynomials but the landmark extraction and data association algorithm has been developed so that the map is formed by both x^1-variate and x^2-variate polynomials.

At each step k, using the robot pose prediction $\hat{x}_{k|k-1}$ and the measurements y_k provided by the sensors, the landmark extraction and data association process modifies the polynomials set $\mathbb{E}_{k-1}$ into $\mathbb{E}_k$. First, starting from $\hat{x}_{k|k-1}$ and y_k, points $\pi_k^i, i = 1, \ldots, n_S$ are obtained as

$$\pi_k^i = \begin{bmatrix} \hat{x}^1_{k|k-1} + y_{i,k} \cos(\hat{\theta}_{k|k-1} + \beta_i) \\ \hat{x}^2_{k|k-1} + y_{i,k} \sin(\hat{\theta}_{k|k-1} + \beta_i) \end{bmatrix} \tag{3}$$

where $y_{i,k}$ is the measurement provided by sensor S_i at step k and β_i is the i-th sensor heading w.r.t the robot axis. These points can be seen as an approximation of the points $\tilde{Q}_i$ (see Fig. 1) due to the estimation errors and measurements errors.

For each point π_k^i, the data association process is performed. As a first step, among the polynomials in $\mathbb{E}_{k-1}$, the one best approximating π_k^i is found. Given a polynomial segment p, denote by Q_m and Q_M its starting and ending points: if a point π is in the region approximated by p, then the approximation error is the LMS error due to the use of p in modeling π; otherwise we define the approximation error as $\min(||Q_m - \pi||^2, ||Q_M - \pi||^2)$: see Fig. 2. Let p_k^* be the polynomial best approximating π_k^i and let ε_k be the related approximation error. The error is checked and three cases may occur.

The first one is related to a good approximation, that is $\varepsilon_k \leq \rho_k$, where $\rho_k > 0$ is a given threshold; in this case no further actions are performed.

The second case is $\rho_k < \varepsilon_k \leq \sigma_k$, where σ_k is a given threshold; here, p_k^* should be used to approximate π_k^i but it is not good enough in modeling this point. The point π_k^i is then stored into the set $\mathcal{Q}_{bad}(p_k^*)$, which contains the points that should be approximated by p_k^* but their related approximation error is too high.

The last case is related to $\sigma_k < \varepsilon_k$; in this situation the approximation error due to the use of p_k^* in modeling π_k^i is too high and the point is considered to be related to an environment portion not modeled into the map yet. A set of points clusters, $\mathcal{Z}_{k-1} = \{Z_j, j = 1, \ldots, n_z\}$ is used and each cluster contains a set of environment boundaries points which have not been mapped yet. Point π_k^i is then inserted into the cluster $Z_j \in \mathcal{Z}_{k-1}$ containing the largest number of points neighbors, in a radius R, to π_k^i. If there is no cluster that contains at least a point neighbor to π_k^i, a new cluster is created containing only this point.

As for the ρ_k and σ_k thresholds, they are set at each step considering the current estimation error: the bigger the error is, the higher each threshold has to be set, in order to not refuse correct approximation situations due to an unreliable state estimation. A possible way to set them is

$$\begin{array}{l} \sigma_k = \underline{\sigma} + (\overline{\sigma} - \underline{\sigma})(||KP_0|| - ||KP_0 - P_{k|k-1}||)/||KP_0|| \\ \rho_k = \underline{\rho} + (\overline{\rho} - \underline{\rho})(||KP_0|| - ||KP_0 - P_{k|k-1}||)/||KP_0|| \end{array},$$

where $\underline{\sigma}, \overline{\sigma}$ $(\rho, \overline{\rho})$ are the thresholds minimum and maximum allowed values, respectively, P_0 is the initial prediction error covariance matrix value and $K > 0$ is one of the algorithm parameters.

The main idea behind the above equation is to use a threshold linearly varying w.r.t. the prediction error covariance matrix $P_{k|k-1}$ in a range $\emptyset \leq P_{k|k-1} \leq KP_0$. If $P_{k|k-1} > KP_0$ then the prediction error covariance matrix is too large and the filter is considered diverging.

Let now consider the case where $\rho_k < \varepsilon_k \leq \sigma_k$; if after including the point π_k^i within $\mathcal{Q}_{bad}(p_k^*)$ the number of points in the above set becomes larger than a threshold ε_M, then p_k^* is modified to better adapt it to the badly approximated points. To this end, a set $\mathcal{R}$ of ε_M equally spaced points on p_k^*, is computed. A new polynomial p is now computed by minimizing the least mean square error due to the use of a m-th order polynomial to approximate the points contained in $\mathcal{Q}_{bad}(p_k^*) \bigcup \mathcal{R}$. The new polynomial p will be used to map the environment and will substitute the polynomial p_k^*; the related set of badly approximated points will be an empty set, $\mathcal{Q}_{bad}(p) = \emptyset$.

Where $\varepsilon_k > \sigma_k$, if after including point π_k^i into a cluster Z the number of points in that cluster becomes larger than a threshold ε_Z, then the cluster is removed from $\mathcal{Z}_{k-1}$ and a new polynomial p is computed by finding the best LMS m-th order polynomial approximation for the points in Z, and that polynomial is inserted in the landmarks set $\mathbb{E}_{k-1}$.

When all the currently acquired points $\{\pi_k^i, i = 1, \ldots, n_S\}$, have been used, the obtained landmarks set $\mathbb{E}_{k-1}$ and the clusters set $\mathcal{Z}_{k-1}$ become the updated environment map: $\mathbb{E}_k = \mathbb{E}_{k-1}$ and $\mathcal{Z}_k = \mathcal{Z}_{k-1}$.

The dimensions of the augmented state X_k and of the matrices involved into the filter are time varying, since the number of polynomials used to map the environment changes during the SLAM process. For each change in the landmarks there is a change in the augmented estimation error covariance matrix $\mathcal{P}_k$ and in X_k. More precisely, for each landmark added/removed, the related position coefficient is added/removed from X_k. As for the $\mathcal{P}_k$ matrix, if a landmark is removed from the map, the related rows and columns are removed from the covariance matrix; if a new landmark is inserted into the map, then, as shown in Neira et al (2001), that matrix becomes

$$\tilde{\mathcal{P}}_k = G_x \mathcal{P}_k G_x^T + G_y V G_y^T, \quad \mathcal{P}_k = \tilde{\mathcal{P}}_k, \tag{4}$$

where the G_x and G_y matrices are computed as

$$G_x = \left. \frac{\partial g(x,y)}{\partial x} \right|_{\substack{x = \hat{x}_{k|k-1} \\ y = y_k}}, \; G_y = \left. \frac{\partial g(x,y)}{\partial y} \right|_{\substack{x = \hat{x}_{k|k-1} \\ y = y_k}}, \tag{5}$$

and $g(x, y)$ is the landmark generation function used to obtain the new landmark. Using that update rules the augmented state and the estimation error covariance matrix are properly modified according to the variations in the environment mapping.

3.2 Data exchange process

The second part of a multi-robot SLAM algorithm is related to the interactions of robots once they meet. Let the robots poses be given in the absolute reference frame.

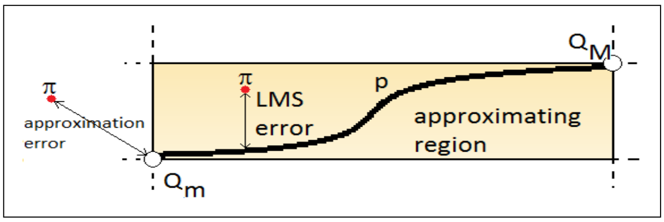

Fig. 2. Polynomial approximating region and approximation errors.

Each robot r_j starts its path from an unknown initial position $x_{j,0}, j = 1, \ldots, N$. As a consequence, each robot assumes to start from $\hat{x}_{j,0} = [0\ 0\ 0]^T$ and uses this pose as its relative mapping and localization reference frame.

Let $R_{j,k}$ be the reference frame consistent with robot r_j at step k: it is centered on robot r_j estimated position $(\hat{x}^1_{j,k}, \hat{x}^2_{j,k})$ and its x^1 axis is rotated of an angle $\hat{\theta}_{j,k}$ w.r.t. the robot relative reference frame $R_{j,0}$. The algorithm outputs will be the robots estimated trajectories and maps, $\hat{x}_{j,k}, \mathbb{E}_{j,k}, j = 1, \ldots, N$, and they will be related to the relative robot reference frame $R_{j,0}$, and thus biased w.r.t. the absolute reference frame by $x_{j,0}$ offsets, respectively.

Each robot starts its path into the environment, taking measurements by its distance sensors and running its own SLAM algorithm. No assumptions are made about robots relative pose; they move as they were on their own and interact only when a rendezvous occurs. A rendezvous happens when two robots r_j and r_l detect each other; in that case, the two robots establish a connection allowing r_j sending its map $\mathbb{E}_{j,\overline{k}}$ to r_l and vice versa. This event is denoted by the involved robots (r_l, r_j) and the time step it occurs, $\overline{k}$. In the following, the index l will denote the robot that sends the data, while the index j will denote the robot that receives data from r_l.

To use the map $\mathbb{E}_{l,\overline{k}}$, received by r_l, robot r_j has to rotate and translate this map w.r.t. its own reference frame $R_{j,0}$, coping with the bias due to the different reference frames. This transformation translates each polynomial in $\mathbb{E}_{l,\overline{k}}$, expressed in $R_{l,0}$, into the reference frame $R_{j,0}$. However, when a polynomial is rotated, the resulting curve is no more a polynomial in the new reference frame, unless $m = 1$. To face this problem, once r_j has received the coefficients of a m-th order polynomial p from r_l, it approximates the polynomial by a piecewise linear approximation made of a set of segments $\{s_\tau, \tau = 1, \ldots, \mathcal{T}\}$.

After the segments $\{s_1, \ldots, s_\mathcal{T}\}$ have been obtained, their starting/ending points are rotated and translated to express them into the r_j reference frame. This transformation is obtained by finding the transformation matrix $T_{l0,j0}$ from the reference frame $R_{l,0}$ to the reference frame $R_{j,0}$. As shown in Carbone et al (2011), the transformation matrix can be found using the rendezvous information:

$$T_{l0,j0} = T_{j,j0}\ T_{l,j}\ T^{-1}_{l,l0},$$

where $T_{l,l0}$ is the transformation matrix from the reference frame $R_{l,\overline{k}}$ consistent with robot r_l during the rendezvous to the reference frame $R_{l,0}$. Along the same lines, $T_{j,j0}$ is the transformation matrix from the reference frame $R_{j,\overline{k}}$ to the reference frame $R_{j,0}$. The transformation matrices are found using the information about $\hat{x}_{l,k|k}$ and $\hat{x}_{j,k|k}$, respectively. Finally, $T_{l,j}$ is the transformation matrix from $R_{l,\overline{k}}$ to $R_{j,\overline{k}}$, and it depends on the relative pose between robots:

$$T_{l,j} = \begin{bmatrix} \cos(\vartheta_{l,j}) & -\sin(\vartheta_{l,j}) & d_{l,j}\cos(\vartheta_{l,j}) \\ \sin(\vartheta_{l,j}) & \cos(\vartheta_{l,j}) & d_{l,j}\sin(\vartheta_{l,j}) \\ 0 & 0 & 1 \end{bmatrix},$$

where $d_{l,j}$ is the relative distance between robots, and $\vartheta_{l,j} = \pi + \alpha_{l,j} - \alpha_{j,l}$; angles $\alpha_{l,j}$ and $\alpha_{j,l}$ are shown in Fig. 4. Using $T_{l0,j0}$, the map $\mathbb{E}_{l,\overline{k}}$ is transformed into a map in the r_j reference frame. Let this map be $\mathbb{E}_{l\rightarrow j,\overline{k}}$,: robot r_j can now update its own map by adding to it $\mathbb{E}_{l\rightarrow j,\overline{k}}$.

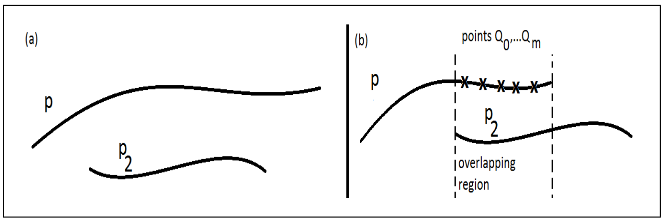

Fig. 3. Landmarks merging situations: (a) p_2 included into p; (b) polynomials have an overlapping region.

Remark 1 The solution here proposed only requires to transfer the environment modeling polynomials from the previous rendezvous to the current one, thus the amount of data to be transferred is extremely low. Moreover, if two robots meet more than one time, they have to exchange only the polynomials extracted after the last rendezvous.

Remark 2 The proposed strategy uses polynomials to model the environment and then approximates them as a set of segments when they have to be rotated. An alternative approach could be the direct use of segments, as shown in Castellanos et al (2007). However, using polynomials instead of lines is expected to yield to better mapping results, and if the segments based approximation before the rotation is sufficiently accurate, the resulting set of rotated segments should preserve the accuracy due to the polynomial approximation.

3.3 Map merging

Map merging is a problem of interest for both the single robot and the multi-robot SLAM context, because in both cases, when a new landmark p is obtained from another robot or it is extracted from current measurements, it has to be compared to previously acquired ones and eventually merged with one of them. Following the same lines as in Pedraza et al (2009), the map merging process is based on the geometrical analysis of the polynomial landmarks.

Two cases may occur: in the first case, the extrema of p are included or completely include the extrema of a mapping landmark p_2: see Fig. 3(a); in the second case, the two polynomials p and p_2 overlap for a sufficiently large region: see Fig. 3(b). In both cases, let $Q_0, Q_1, \ldots, Q_m$ be $m+1$ equally spaced points on p in the overlapping region between p and p_2: see Fig. 3(b). Let γ be the LMS approximation error due to using p_2 in mapping the Q_i points. If γ is lower than a threshold γ_M, then p and p_2 are merged by creating a new LMS polynomial from ε_M points equally spaced on p and ε_M points equally spaced on p_2.

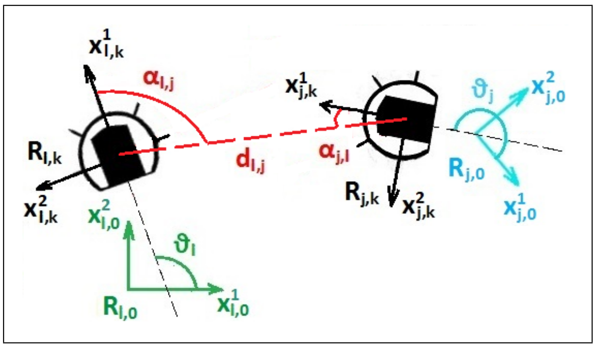

Fig. 4. Reference frames involved in coordinate transformation during a rendezvous event.

To summarize, three landmark creation situations may occur: (1) a new landmark is created from measurements not mapped yet; (2) an existing landmark is modified because of a high number of badly approximated measurements; (3) a new polynomial is computed by merging two existing ones. In all the cases, a weighted LMS technique is used. The badly approximated points and the not mapped points are weighted by $1/||P_{k|k-1}||$, where $P_{k|k-1}$ is the covariance matrix related to the predicted pose used to compute that point. The points in $\mathcal{R}$ are weighted using the inverse covariance related to the innovation landmark relative to the polynomial to be updated; the same technique is used in the third case for the polynomials to be merged. As a consequence, the function $g(x, y)$ of equations (5) is the weighted least mean square function: depending on the landmark creation, its derivatives are computed taking into account polynomials rotation, translation and update.

4. NUMERICAL RESULTS

To assess the performance of the proposed algorithm, numerical simulations have been performed using $N = 3$ differential drive robots, each one assumed to be equipped with five distance sensors to detect the environment and one camera to ensure other robots detection.

A set of 100 numerical simulations have been performed with the following parameters: sampling period $T = 1$ s, $m = 3$, $\underline{\rho} = 0.08$ m, $\overline{\rho} = 0.24$ m, $\underline{\sigma} = 0.1$ m, $\overline{\sigma}_m = 0.3$ m, $\varepsilon_M = 5$, $\varepsilon_Z = 10$, $K = 150$, $R = 0.35$ m, $\gamma_M = 0.35$ m. Distance sensors are assumed to be affected by a Gaussian noise with zero mean and standard deviation of 0.02 m. The process noise is assumed to have a standard deviation of 0.001 m on the position and of 0.01 degrees on the heading. A matrix $P_0 = \text{diag}\{0.05^2, 0.05^2, 0.0175^2\}$ has been used as initial estimation error covariance matrix for all the robots.

The robots move in the environment shown in Figs. 5, 6, 7 and their angular velocities have been precomputed to make them follow the shown paths. Each robot moves in only a part of the environment and could not map all the environment boundaries on its own. Each path has been performed in $k_f = 200$ steps ensuring rendezvous only between robots r_2 and r_3. To evaluate the algorithm localization performance, a NEES index (Bar-Shalom et al (2001)) has been used:

$$\mu_j = \frac{1}{k_f+1}\sum_{k=0}^{k_f}\left[x_{j,k} - \tilde{x}_{j,k|k}\right]^T P_{j,k|k}\left[x_{j,k} - \tilde{x}_{j,k|k}\right]$$

Table 1. Averaged indexes over 100 simulations

	robot r_1	robot r_2	robot r_3
μ_j	$1.66 \cdot 10^{-6}$	$4.66 \cdot 10^{-7}$	$1.02 \cdot 10^{-6}$
ν_j	0.17 m	0.19 m	0.34 m
τ_j	0.61 s	0.66 s	0.62 s

where $P_{j,k|k}$ is the robot r_j localization error covariance matrix and $\tilde{x}_{j,k|k}$ is the robot r_j estimated pose in the absolute reference frame (compensating the bias due to the use of $\hat{x}_{j,0} = [0\ 0\ 0]^T$).

The mapping performance has been evaluated through a further index ν_j. The proposed SLAM algorithm provides a set of mapping polynomials for each of the involved robots: $\mathbb{E}_j = \{p^q_j\}^{n_j}_{q=1}$. A curvilinear abscissa is defined on each polynomial and a set of equally spaced points is taken on the polynomial shape using a step of 0.01. For each point $Q^i_{q,j}$ on polynomial p^q_j extracted by robot r_j, the distance $\delta^i_{q,j}$ between this point and its nearest real environment boundaries point is computed, and used to obtain the overall mapping error:

$$\nu_j = \frac{1}{n_j}\sum_{q=1}^{n_j}\nu_{q,j}, \text{ where } \nu_{q,j} = \sum_i \delta^i_{q,j}.$$

Averaged results for μ_j, ν_j and the computation time [2] τ_j, over the 100 simulations are shown in Table 1.

Typical results of the proposed SLAM algorithm are shown in Figs. 5, 6, 7 for the three robots. Fig. 8 shows the results of a single robot SLAM algorithm for robot r_3. Due to the chosen trajectories, robot r_1 is not involved in any rendezvous event and, as shown in Fig. 5, its map is incomplete and only related to a part of the environment. Moreover, using a multi-robot approach, the environment parts detected by both the robots involved into a rendezvous event are more accurately mapped w.r.t. the single robot case, thanks to the use of more measurements acquired in different locations; see the left up environment part in Figs. 7 and 8.

Table 1 shows that both the error due to robots pose localization and the one related to environment mapping are very low, thus the proposed algorithm is efficient in solving the SLAM problem. Moreover, the computation times are sufficiently low w.r.t. the chosen sampling period to use the algorithm in real time during robots motion.

5. CONCLUSIONS

A novel solution to the Simultaneous Localization and Mapping problem for a team of mobile robots has been proposed. The algorithm is based on approximating the environment by a set of polynomials, the use of which ensures high mapping performance. Also, when two robots are involved into a rendezvous, only the acquired polynomials data have to be exchanged, with a low communication effort. Once a robot has received the information from another one, it transforms this information in its own reference frame by means of simple geometric considerations.

[2] The algorithm computation times have been computed using Matlab R2012 running on an Intel(R) Core(TM) i7 Q720 CPU

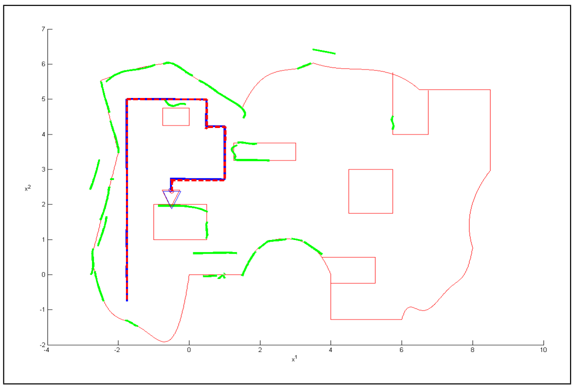

Fig. 5. Multi-robot SLAM result for robot r_1; green: mapped environment; red: real environment; blue: real pose; dashed red: estimated pose.

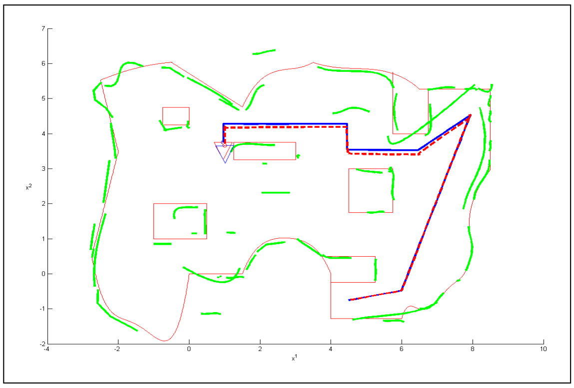

Fig. 6. Multi-robot SLAM result for robot r_2; green: mapped environment; red: real environment; blue: real pose; dashed red: estimated pose.

Numerical simulations have shown the effectiveness of the proposed algorithm both in mapping the environment and in localizing the robots.

Experimental tests using a team of Khepera III mobile robots (by K-TEAM Corporation) in a real environment are in progress to further validate the proposed technique.

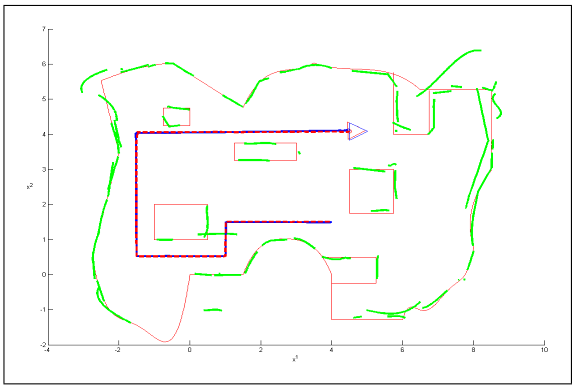

Fig. 7. Multi-robot SLAM result for robot r_3; green: mapped environment; red: real environment; blue: real pose; dashed red: estimated pose.

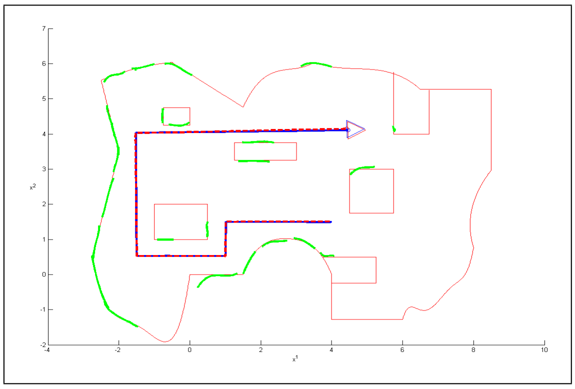

Fig. 8. Single robot SLAM result for robot r_3.

REFERENCES

Smith, R. C., and Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *Int. J. Robot. Res.*, 5(4), 56–68.

Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., and Stewart, B. (2006). Distributed multirobot exploration and mapping. *Proc. IEEE*, 94(7), 1325–1339.

Thrun, S. (2001). A probabilistic online mapping algorithm for team of mobile robots. *Int. J. Robot. Res.*, 20(5), 335–363.

Howard, A., Sukhatme, G., and Mataric, M. J. (2006). Multirobot simultaneous localization and mapping using manifold representations. *Proc. IEEE*, 94(7), 1360–1369.

Zhou, X. S., and Roumeliotis, S. I. (2006) Multi-robot SLAM with unknown initial correspondence: the robot rendezvous case. *Proc. RSJ Conf.*, Beijing, China, Oct. 9-15, 2006, pp. 1785–1792.

Tong, T., Yalou, H., Jing, Y., and Fengchi, S. (2008). Multi-robot cooperative map building in unknown environment considering estimation uncertainty. *Proc. CCDC Conf.*, Yantai, Shandong, China, July 2-4, 2008, pp. 2896–2901.

Carbone, L., Kaouk Ng, M., Du, J., Bona, B., and Indri, M. (2011). Simoultaneous localization and mapping using Rao-Blackwellized particle filters in multi robot systems. *J. Intell. Robot. Syst.*, 63(2), 283–307.

D'Alfonso, L., Grano, A., Muraca, P., and Pugliese, P. (2013). A polynomial based SLAM algorithm for mobile robots using ultrasonic sensors - Experimental results. *Proc. ICRA Conf.*, Montevideo, Uruguay, November 25-29, paper #0111.

Neira, J., and Tardòs, D. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE T. Robot. Autom.*, 17(6), 890–897.

Castellanos, J. A., Martinez-Cantin, R., Tardòs, J. D., and Neira, J. (2007). Robocentric map joining: Improving the consistency of EKF-SLAM, *Robot. Auton. Syst.*, 55(1), 21–29.

Pedraza, L., Rodriguez-Losada, D., Matia, F., Dissanayake, G., and Miro, J.V. (2009). Extending the limits of feature-based SLAM with B-splines. *IEEE T. Robot.* 25(2), 353–366.

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2001). *Estimation with application to tracking and navigation.* John Wiley and Sons.