Switching Multirobot Collaborative Localization in Symmetrical Environments

F. Abrate, B. Bona, M. Indri, S. Rosa and F. Tibaldi

***Abstract*— This paper addresses and solves the problem of multirobot collaborative localization in highly symmetrical 2D environments. Symmetrical environments can be encountered in logistic application scenarios, where a team of rovers moves along several parallel corridors in a large surface, to perform surveillance and monitoring tasks. Because of the environment symmetry, current algorithms fail to provide a correct estimate of the position and orientation of the rover, if its initial position is not known, no specific landmark is introduced, and no absolute information (e.g., GPS) is available: the rover can estimate its position with respect to the walls of the corridor, but it could not determine in which corridor it is actually moving. The proposed algorithm is based upon a particle filter cooperative Montecarlo Localization (MCL), as in [2], and implements a two-stage procedure that guarantees global localization as well as position tracking of each rover in a team. The simulation tests, which investigate different numbers of involved rovers, their initial positions, and some possible critical situations, show how the proposed solution can lead to the global localization of each rover, with a precision sufficient to be used as starting point for the subsequent rover tracking.**

I. Introduction

The multirobot case is one of the most challenging and promising areas in the current and future mobile robotics research, since a coordinated team of rovers can be successfully employed in various application scenarios, e.g., for surveillance and monitoring tasks in different fields.

Independently of the particular application, the correct rover localization is always required and it consequently constitutes one of the most fundamental problems in mobile robotics: [15] offers a comprehensive study of such a problem. The multirobot case potentially gives some interesting advantages, since the accuracy of the rovers position estimates can be improved by a cooperative localization, even if communication and data sharing problems must be taken into account. The most common approaches are based on Extended Kalman Filters (EKF) methods or on Monte Carlo Localization (MCL) methods. In the EKF approaches (see e.g., [5], [6], [8], [9], [11], [13]) the data association problem is generally solved by using multi modal distributions, instead of a single Gaussian one, to approximate the position probability distribution, sometimes including iterations that propagate also an approximation of the posterior marginal densities of the unknown variances. In the MCL methods (see e.g., [2], [4], [10], [12]) an arbitrary posterior probability distribution is considered by using particle filters. Cooperative localization approaches based on robust estimation techniques, in which unknown but bounded error models are employed for the sensor measurements, have also been proposed, e.g., in [7], [14].

Two sub-problems can be distinguished within the localization framework: *position tracking* and *global localization*. In the first one, the rover pose must be iteratively estimated starting from an initial condition known with a given uncertainty, while in the second one (the most challenging in general) the correct global rover position with respect to the environment map must be determined without any information about its initial value, as well as starting from a completely wrong estimation of its initial pose as in the so-called *kidnapped robot* problem.

Many works available in literature use multirobot and/or mutual localization to improve the quality of the self-localization results that each single rover could achieve on the basis of its own sensors only, implicitly assuming that the information provided by such sensors would be sufficient to obtain a *macroscopically* correct global localization, even if not precise. Such an assumption does not hold when the environment in which the rovers are moving is completely symmetrical: in this case a correct global self-localization cannot be performed by the single rover without any external help.

Symmetrical environments can be encountered quite easily in some application scenarios, like the one considered in this paper, which deals with a team of rovers moving in a large logistic space to perform surveillance and monitoring tasks. The considered logistic space is intended as an indoor or outdoor area, where logistic and transport societies receive and store large quantities of goods, mainly bulky ones, as containers, cars, and other similar items. In order to achieve an efficient occupancy of the area and facilitate the transport operations, the free corridors of the area among the stored goods form a regular grid, as in Figure 4. The resulting symmetry of the environment map precludes the correct global self-localization of each rover, if its initial position is not known, no specific landmark is introduced to distinguish each corridor, and no external information (like GPS) is available or exchanged with the other rovers: by using only its own sensors (odometry, laser scanner, sonar, etc.), each rover could well estimate its position with respect to the walls of the corridor, but it could not determine in which corridor it is actually moving.

This paper investigates the possibility of using a cooperative MCL approach (as in [2]) to solve such a global

This work was supported under the MACP4Log grant (RU/02/26) from Regione Piemonte.

F. Abrate, B. Bona, M. Indri, S. Rosa, F. Tibaldi are with Department of Control and Computer Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy fabrizio.abrate@polito.it

localization problem, without forcing artificial asymmetries in the environment. The proposed solution does not use any external information (e.g., GPS), that could be unavailable in some indoor areas, but occasional noisy measurements coming from a compass sensor are sufficient to resolve position ambiguity due to symmetry. The reported simulation tests, which investigate different cases with respect to the number of involved rovers, their initial positions, and some possible critical situations, show how the proposed solution can lead to the global localization of each rover, with a precision sufficient to be used as starting point for the subsequent rover tracking, within a two-stage localization procedure.

The paper is organized as follows: Section II describes the proposed localization algorithm called **SMCL**; Section III is devoted to the tests designed and performed to demonstrate the effectiveness of the proposed algorithm; Section IV draws some final conclusions and discusses future works.

II. The Switching Multirobot Collaborative Localization (SMCL)

A. Preliminaries

The ***S**witching **M**ultirobot **C**ollaborative **L**ocalization* algorithm (**SMCL**) allows each member of a group of rovers moving in a highly symmetrical area (e.g., a large logistic space) to accurately localize itself and to correctly track its position over time. The **SMCL** algorithm correctly operates for rovers endowed with at least sonar range sensors, a monocular or an omnivision camera and a compass sensor. The camera is used to detect the positions of other rovers when they are in the field of view; the measurement precision may be improved using a laser range finder if available. A binary occupancy grid map of the environment is assumed to be available.

The algorithm first performs *global localization* over the map in a decentralized way, exploiting the position estimates of the other rovers of the group, then realizes when the localization error estimate is lower than a given threshold, causing to switch to a pure *position tracking* algorithm. Finally, the algorithm allows rovers to detect a sudden increase in the localization error, due for instance to kidnapping or failures in proprioceptive sensors (e.g., wheel encoders rupture). In this case the algorithm switches again to *global localization*.

Let $\mathcal{R} = \{r_i : i = 1, \ldots, N_R\}$ be the set of rovers deployed in the area; with t we indicate the time variable that clocks the whole localization algorithm. With $d_i(t)$ we indicate data coming form the i-th robot proprioceptive and exteroceptive sensors at time t. In particular we have that

$$d_i(t) = \begin{cases} o_i(t) & \text{if proprioceptive measurement} \\ z_i(t) & \text{if exteroceptive measurement} \end{cases}$$

The proprioceptive measurement $o_i(t)$ is used to perform dead-reckoning, while the exteroceptive measurement $z_i(t)$ contains the range measurements given by the range sensors.

Each rover is able (a) to measure the positions of the other rovers in the field of view of its vision sensor in its local reference frame, concurrently with the **SMCL** algorithm, (b) to transform the measurements in a global reference frame common to all rovers, and (c) to finally send these values to the detected rovers via a wireless link.

Let k denote a time instant at which the position of the i-th rover is detected by a set of rovers $R_i(k) \subseteq R$, ($|R_i(k)|$ being its cardinality). The rovers belonging to $R_i(k)$ send their measurements to the i-th rover, which collects them in the following vector:

$$h_i(k) = \begin{bmatrix} \hat{x_i}^1(k), \hat{y_i}^1(k) \\ \vdots \\ \hat{x}_i^{|R_i(k)|}(k), \hat{y}_i^{|R_i(k)|}(k) \end{bmatrix}. \quad (1)$$

Each row of (1) contains an hypothesis on the position of the i-th rover expressed in Cartesian global coordinates.

The set of all the measurements received by the i-th rover up to time k is then defined as $H_i^k = \{h_i(1), \ldots, h_i(k)\}$.

B. The algorithm

We now describe the core of the **SMCL** algorithm, which runs onboard each rover and it is outlined in Algorithm 1.

Input: χ^{t-1}, $d_i(t)$, $R_i(k)$, H_i^k, N_{min}, N_{max}, N_{hyp}, m
Output: $\Phi_i(t)$, $\phi_i^{best}(t)$
if *flag = 1* **then** Position Tracking
  [χ^t,μ_k,l] = **position_tracking**($d_i(t)$, χ^{t-1}, $h_i(k)$, l)
  [$\Phi_i(t)$, $\phi_i^{best}(t)$]= **DT_clustering**(χ_t)
  if $l > n_{pt}$ **then**
    [μ_k] = **loc_perf**($\phi_i^{best}(t)$, $h_i(k)$)
    if $\mu_k \geq \mu_{thu}$ **then**
      $flag = 0$; $l = 0$
    end
  end
else Global Localization
  initialize χ_t;
  if $d_i(t)$ = $o_i(t)$ **then**
    $p_i(t)$ = **sample_motion_model**($d_i(t)$, $p_i(t-1)$);
  else if $d_i(t)$ = $z_i(t)$ **then**
    $w_i(t)$ = **measurement_model**($d_i(t)$, $p_i(t)$, m)
    $\bar{\chi}^t$ = $\bar{\chi}^t + \langle p_i(t), w_i(t) \rangle$
    if $R_i(k) = \emptyset$ **then**
      χ^t = **KLD_1**($\bar{\chi}^t$, N_{min}, N_{max})
      [$\Phi_i(t)$, $\phi_i^{best}(t)$] = **DT_clustering**(χ^t) ;
    else
      $l = l + 1$;
      χ^t = **KLD_2**($\bar{\chi}^t$, N_{min}, N'_{max}, N_{hyp}, $h_i(k)$)
      [$\Phi_i(t)$, $\phi_i^{best}(t)$] = **DT_clustering**(χ^t)
      if $l > n_{gl}$ **then**
        [μ_k] = **loc_perf**($\phi_i^{best}(t)$, H_i^k)
        if $\mu_k \leq \mu_{thd}$ **then**
          $flag = 1$; $l = 0$;
        end
      end
    end
  end
end

Algorithm 1: The SMCL algorithm in pseudocode

The algorithm is basically organized in two stages. The first stage is active when the rover performs *global localization* (lines 12-38), while the second stage is active when the rover performs *position tracking* (lines 1-10). At the beginning it always enters the first stage, and switches to the second stage when the localization performances are sufficiently accurate. Similarly, when the rover is in *position tracking*, it continues to monitor its localization performances. In case of localization performance degradation, the algorithm switches again to *global localization*.

Both the stages are based on particle filters [15]; observing the *global localization* pseudo-code in Algorithm 1, it can be noticed the typical prediction phase at line 15 and the update phase at lines 17-18. The prediction phase computes the vector $p_i(t)$ containing the predicted pose (in terms of global coordinates $\{x,\ y,\ \theta\}$) for each particle, while the purpose of the update phase is twofold. It gives the vector $w_i(t)$ containing the importance factors for each particle, and it verifies if vector $h_i(k)$ contains position estimates outside the map. If this is the case, such estimates are weighted using a Bivariate Normal Distribution. Then, at line 19, the algorithm verifies if it has received a vector of measurements $h_i(k)$ from other rovers of the set $R_i(k)$ at time k. If $R_i(k)$ is empty, a classic *Kullaback-Leibler Divergence* (KLD) Resampling occurs (see [15]); N_{min} and N_{max} initialized at line 13 are respectively the lower and upper bound of the number of particles N_{kld} employed in the resampling algorithm. If instead $R_i(k)$ is not empty, a modified version of the KLD Resampling has been implemented (line 24). The idea is to exploit the relative Cartesian position measurements (contained in vector $h_i(k)$) that the i-th rover receives from the other rovers of $R_i(k)$ to remove the ambiguity on localization due to the symmetry of the environment. To achieve this goal, the algorithm distributes a subset N'_{kld} of the resampled particle set N_{kld} around the elements of the vector $h_i(k)$. A new bound N'_{max} on the number of particles N_{kld} is set as:

$$N'_{max} = N_{max} - N_{hyp},$$

where N_{hyp} is the minimum number of particles that can be Gaussianly distributed around the elements of $h_i(k)$, and hence $N'_{kld} \geq N_{hyp}$. Therefore it holds that $N_{min} \leq N_{kld} \leq N'_{max}$.

After the resampling phase, a classic *Density-Tree* clustering [2] (lines 3, 21, 25) is always performed, that provides a vector of hypotheses $\Phi_i(t)$ on the position of the i-th rover and the best hypothesis $\phi_{best}(t)$. The vector of hypotheses is defined as

$$\Phi_i(t) \quad = \quad \begin{bmatrix} p_i^1(t), \Sigma_i^1(t), W_i^1(t) \\ \vdots \\ p_i^{|\Phi_i(t)|}(t), \Sigma_i^{|\Phi_i(t)|}(t), W_i^{|\Phi_i(t)|}(t) \end{bmatrix},$$

where $|\Phi_i(t)|$ is the cardinality of $\Phi_i(t)$, Σ_i^j are the covariance matrices relative to the the poses $p_i^j(t)$, and finally $W_i^j(t)$ are the weights associated to each hypothesis, representing their level of confidence, with $j = 1, \dots, |\Phi_i(t)|$.

The best hypothesis at time t is defined as

$$\phi_i^{best}(t) = \max_{W_i^j}(\Phi_i(t)) = [p_i^{best}(t), \Sigma_i^{best}(t), W_i^{best}(t)] \quad (2)$$

Switching between position tracking and global localization is based on the following *accordance* function:

$$\mu_k = \sum_{q=k-n}^{k} \sum_{j=1}^{|R_i(q)|} \frac{\sqrt{(\hat{x}_i^j(q) - \hat{x}_i^{best}(t))^2 + (\hat{y}_i^j(q) - \hat{y}_i^{best}(t))^2}}{n|R_i(q)|}, \quad (3)$$

where n is the length of the sliding window used to compute the average of (3), and it is equal to n_{gl} if the rover is in *global localization* and to n_{pt} if the rover is in *position tracking*. The inner summation of (3) averages the distances among the elements of $h_i(k)$ and the best position hypotheses of the i-th rover. The outer summation of (3) performs a moving average of length n on the results of the inner summation. Therefore μ_k measures the accordance between the actual belief on the position of the i-th rover and the average of the beliefs that the other rovers have on its position at time k.

When μ_k is lower than a certain threshold μ_{thd} (empirically determined) the algorithm switches to *position tracking*. This phase is aimed to track the position of the rover over time, and it is implemented in a classic way (see [15]). μ_k is computed also during the position tracking phase: if μ_k becomes greater than a given threshold μ_{thu}, the algorithm switches again to *global localization*.

III. Simulation tests and Results

In this section we demonstrate the effectiveness of the proposed **SMCL** algorithm, carrying out a series of localization experiments in simulation.

The software used to simulate the rovers and their environment is called *MobileSim* [1]. It is based on the *Stage* library [3], and it simulates MobileRobots platforms. We perform experiments with a team of simulated Pioneer 3 DX rovers, endowed with sonar sensors. The simulator embeds a model of the behavior of sonar range sensors, provides rover odometry pose estimation with cumulative error, and allows multiple rovers simulation.

The simulator has also been improved by adding a simple simulated vision sensor, a compass sensor, and the support for communication among rovers.

We consider a simulated environment of a large logistic area (see Figure 4). The occupied black areas can be thought to represent containers or similar bulky items stored by transport societies before distribution. The dimension of the whole environment is 80×65 m, the black areas are 20×10 m and the corridors are 5 m wide.

The symmetry of the environment makes global localization a really difficult task, which the proposed **SMCL** algorithm successfully performs, as the following experiments show in different situations.

A. Experiment 1

The first experiment demonstrates the trend of the localization error for a team of rovers moving in a symmetrical environment and communicating their relative positions and absolute heading - if available - when they are in the field of view. We define the following quantities:

$$e_i^\rho(t) = \sqrt{(x_i^{gt}(t) - \hat{x}_i^{best}(t))^2 + (y_i^{gt}(t) - \hat{y}_i^{best}(t))^2} \quad (4)$$

$$e_i^\theta(t) = \theta_i^{gt}(t) - \hat{\theta}_i^{best}(t) \quad (5)$$

where $e_i^\rho(t)$ is the distance between the ground-truth Cartesian position of the i-th rover $(x_i^{gt}(t), y_i^{gt}(t))$ and its Cartesian position estimation given by the best hypothesis, while $e_i^\theta(t)$ is the difference between the real heading and the actual heading of the best hypothesis. The pose informations of the best hypothesis are given by $p_i^{best}(t)$ and can be extracted by $\phi_i^{best}(t)$, defined in (2).
The experiment consists in initializing randomly the pose of $N_R = 6$ rovers in free areas of the map, let them move according to a simple obstacle avoidance behavior, and monitoring the localization errors $e_i^\rho(t)$ and $e_i^\theta(t)$ for $i = 1, \ldots, N_R$ up to $t = 1600$ s. The results are shown

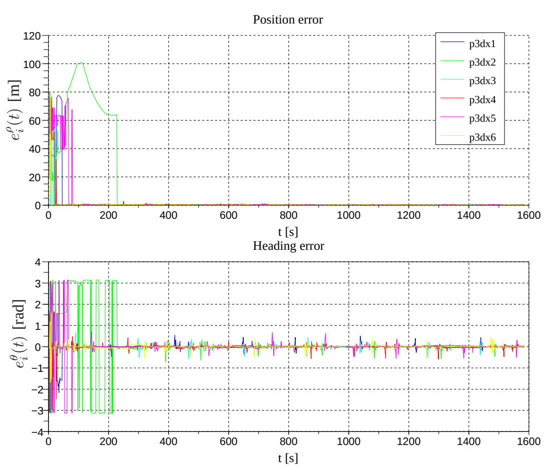

Fig. 1. Experiment 1: localization errors.

in Figure 1. In this experiment all the rovers are correctly localized after approximately 220 seconds, with a remaining position error approximately equal to 30 cm. As soon as *global localization* occurs, the **SMCL** algorithm running on each rover switches to *position tracking*. This can be noticed observing the plots relative to the rover *p3dx2*, whose position and heading errors suddenly decrease at the same time.

B. Experiment 2

In this experiment we analyze the robustness of the **SMCL** algorithm with respect to random variations in the initial position of the rovers. We consider again a team of $N_R = 6$ rovers and we repeat 50 times the experiment described in the subsection III-A, each time setting randomly the initial position of the rovers. Since we are interested in evaluating the average localization error among the repetitions of the experiments, we define $\bar{e}_i^\rho(t)$ as the average of $e_i^\rho(t)$ for the i-th rover over 50 realizations. The results are shown in Figure 2. The localization error

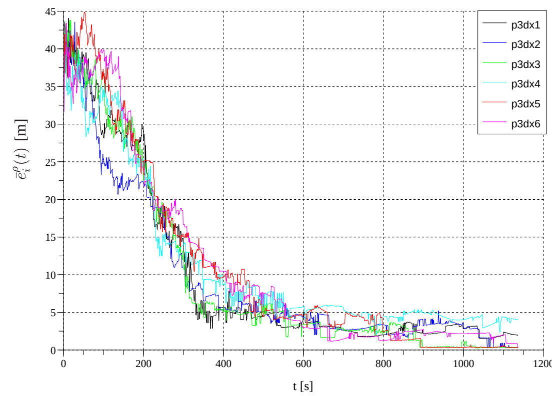

Fig. 2. Experiment 2: average localization errors.

$e_i^\rho(t)$, $i = 1, \ldots, N_R$ decreases approximately linearly for all the rovers; the remaining error of about 5 m would further decrease extending the simulation time up to 1600s, as in Experiment 1. The **SMCL** algorithm is not susceptible to variations in the initial positions of the rovers, even if the environment is symmetrical. This fact has an important impact on the application side, in particular when considering robotic applications in logistic spaces, since the algorithm does not require any particular initial formation of the rovers, avoiding any human intervention to initially place the rovers in a specific area of interest.

C. Experiment 3

In this experiment the same situation of Experiment 1 is considered, but at a certain time a correctly localized rover of the team is kidnapped and moved to another part of the logistic area. This experiment models a set of realistic situations that may lead a rover to lose its localization informations (e.g., due to intermittent failures in the wheels encoders). In Figure 3 the plots of $e_i^\rho(t)$ and $e_i^\theta(t)$ are shown for $i = 1, \ldots, N_R$, while Figure 4 reports a sequence of images representing what happens in simulation; the rover *p3dx2* is kidnapped at approximately $t = 2000$ s (Figure 4 (b)), and the **SMCL** algorithm localizes again the rover at $t = 2150$ s (Figure 4 (g)). Notice that the incorrect position information of the rover *p3dx2* influences for a while the rover *p3dx5*, as soon as *p3dx2* sees *p3dx5* and communicates its incorrect position hypotheses. However *p3dx5* is able to quickly recover correct localization.

D. Experiment 4

This final experiment has been designed to understand how localization performance of the **SMCL** algorithm is affected by the number of rovers in the team, both in terms

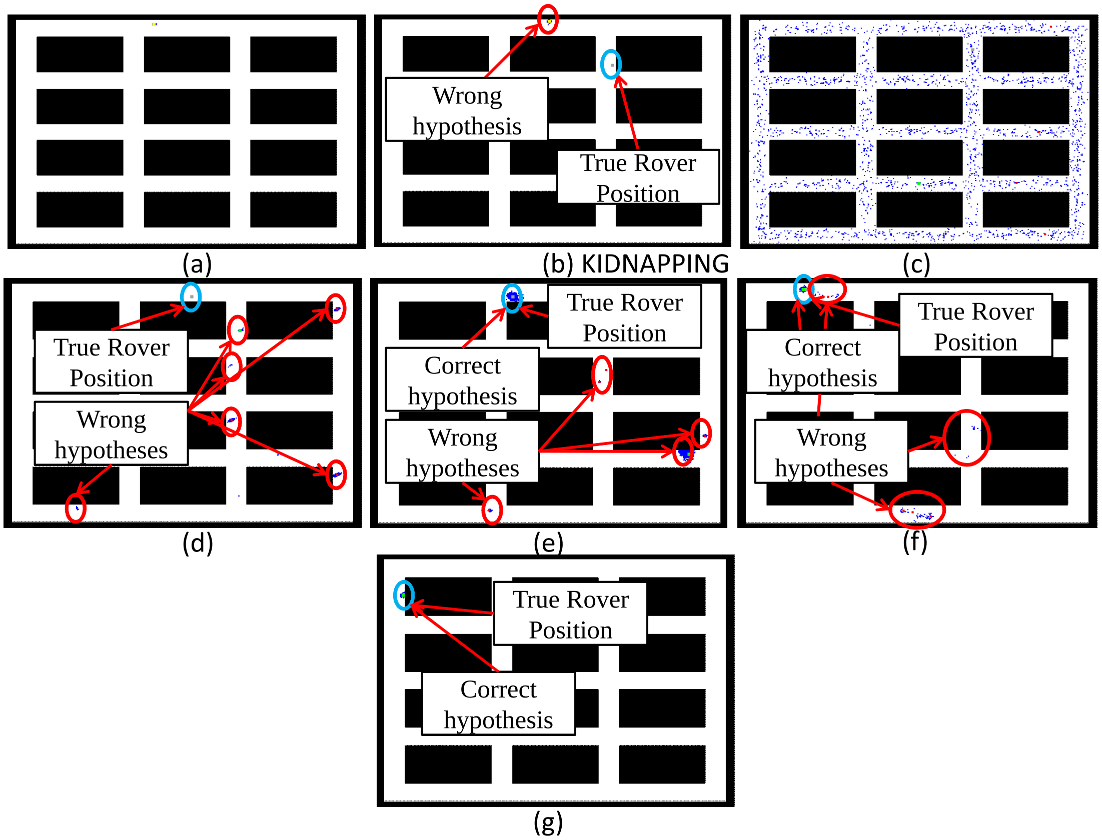

Fig. 4. Experiment 3: snapshots of the kidnap experiment.

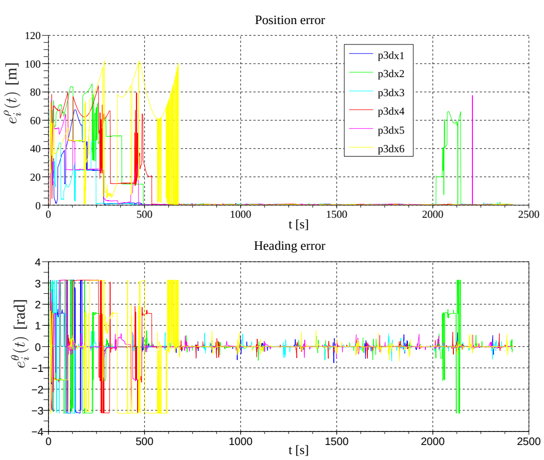

Fig. 3. Experiment 3: localization errors.

of Cartesian position error and of time required to switch to *position tracking*.

We define the average position error among the N_R rovers of the team as:

$$E^{\rho}_{N_R}(t) = \sum_{i=1}^{N_R} \frac{e^{\rho}_i(t)}{N_R} \tag{6}$$

where $e^{\rho}_i(t)$ is the average for the i-th rover over 20 realizations. The results for $N_R = 2, 3, 6, 9$ are reported in Figure 5. It can be clearly seen that two rovers are not sufficient

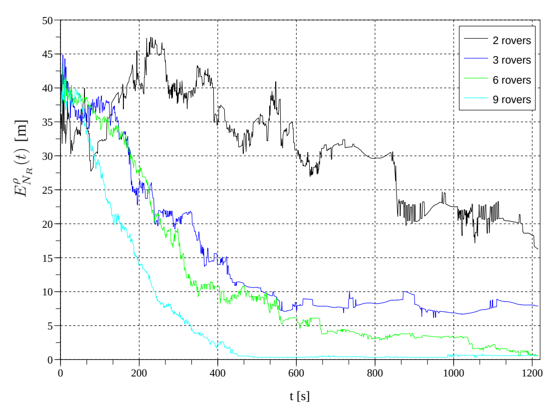

Fig. 5. Experiment 4: average localization errors.

to resolve the ambiguity in localization, but as soon as three rovers are employed, the localization error goes below 10 m after nearly 500 s; increasing the number of rovers up to six and nine strongly improves the performances. In particular with 9 rovers, on average the error becomes lower than about 0.5-0.6 m after $t = 450$ s (see Figure 5). This is particularly important in practical applications, since the localization error decreases and the path planning algorithms become more effective for the rovers relying only on their position estimations, thus allowing rovers to accomplish in a more reliable way the task assigned (e.g., handling hazardous events collaboratively).

The statistics relative to the first switching time to *position tracking* are reported in Figure 6, which shows for each group of N_R rovers the minimum first switching time to *position tracking* among all the realizations of all the rovers (blue column), the average first switching time (green column) and the maximum first switching time (red column). Observing in particular the green columns we can state that *global localization* on average speeds up when the number of rovers is increased.

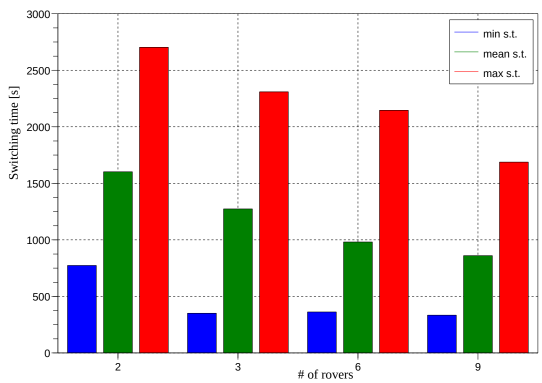

Fig. 6. Experiment 4: switching to *position tracking* times for groups of rovers of increasing cardinality.

IV. Conclusions

The paper has shown how the problem of correct localization of rovers can be successfully solved by the proposed SMCL algorithm, even in a highly symmetrical environment. Thanks to the cooperative action of all the rovers of the team, the approach allows the avoidance of any *ad hoc* intervention to force artificial asymmetries in the environment, and/or the use of coded landmarks to distinguish different regions of the area. The proposed solution can then be usefully adopted in practical applications each time a team of vehicles must autonomously move in an area characterized by a regular grid of corridors or streets. The robustness of the SMCL approach with respect to the initial position of the rovers is an important advantage in practice, since it lets the initial formation of the team be completely arbitrary. Moreover, the automatic switch from position tracking to global localization prevents the occurrence of macroscopical errors, due to temporary failures or rover kidnapping. Finally, it is worth noticing that, even if the performances become better as the number of rovers increases, the tests show that a team of only six rovers can perform the localization task with acceptable results.

Future works will include experimental tests, to confirm in practice the effectiveness of the proposed approach, which has been demonstrated in this paper only by means of simulation tests.

References

[1] "Mobilesim," Website, http://robots.mobilerobots.com/MobileSim/.
[2] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, 2000.
[3] B. Gerkey, R. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *11th Int. Conf. on Advanced Robotics (ICAR 2003)*, 2003, pp. 317–323.
[4] D. Göring and H.-D. Burkhard, "Multi robot object tracking and self localization using visual percept relations," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 31–36.
[5] N. Karam, F. Chausse, R. Aufrere, and R. Chapuis, "Localization of a group of communicating vehicles by state exchange," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 519–524.
[6] A. Martinelli, "Improving the precision on multi robot localization by using a series of filters hierarchically distributed," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007, pp. 1053–1058.
[7] M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino, "Simultaneous localization and map building for a team of cooperating robots: A set membership approach," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 2, pp. 238–249, 2003.
[8] A. Mourikis and S. Roumeliotis, "Performance analysis of multirobot cooperative localization," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 666–681, 2006.
[9] S. Panzieri, F. Pascucci, and R. Setola, "Multirobot localization using interlaced extended kalman filter," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 2816–2821.
[10] M. Peasgood, C. Clark, and J. McPhee, "Localization of multiple robots with simple sensors," in *IEEE Int. Conf. on Mechatronics and Automation*, 2005, pp. 671–676.
[11] G. Pillonetto and S. Carpin, "Multirobot localization with unknown variance parameters," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007, pp. 1709–1714.
[12] I. Rekleitis, G. Dudek, and E. Milios, "Probabilistic cooperative localization and mapping in practice," in *IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 1907–1912.
[13] S. Roumeliotis and G. Bekey, "Distributed multirobot localization," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.
[14] C. Taylor and J. Spletzer, "A bounded uncertainty approach to cooperative localization using relative bearing constraints," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007, pp. 2500–2506.
[15] S. Thrun, W.Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.