# Lidar-Based Relative Position Estimation and Tracking for Multi-Robot Systems

Alicja Wąsik[1,2], Rodrigo Ventura[2], José N. Pereira[1],
Pedro U. Lima[2] and Alcherio Martinoli[1]

[1] Distributed Intelligent Systems and Algorithms Laboratory,
École Polytechnique Fédérale de Lausanne, Switzerland
alicja.wasik@epfl.ch, jose.pereira@epfl.ch, alcherio.martinoli@epfl.ch
[2] Institute for Systems and Robotics,
Instituto Superior Técnico, Universidade de Lisboa, Portugal
rodrigo.ventura@isr.tecnico.ulisboa.pt, pal@isr.tecnico.ulisboa.pt

**Abstract.** Relative positioning systems play a vital role in current multi-robot systems. We present a self-contained detection and tracking approach, where a robot estimates a distance (range) and an angle (bearing) to another robot using measurements extracted from the raw data provided by two laser range finders. We propose a method based on the detection of circular features with least-squares fitting and filtering out outliers using a map-based selection. We improve the estimate of the relative robot position and reduce its uncertainty by feeding measurements into a Kalman filter, resulting in an accurate tracking system. We evaluate the performance of the algorithm in a realistic indoor environment to demonstrate its robustness and reliability.

**Keywords:** Laser Relative Positioning System, Estimation, Tracking

## 1 Introduction

Several applications of cooperative distributed robotic systems (e.g., formation control, coverage) require that each team member is aware not only of its absolute location in a global world frame, but also of the relative locations of its teammates. When wireless communications are reliable each robot can simply regularly share its most recent location estimate with its teammates. Unfortunately, wireless communications are often unreliable, and the amount of communicated data grows quadratically with the number of robots in the team. Thus, a solution not based on communications but relying uniquely on data locally acquired by each robot using its own sensors is desirable.

Position estimation and tracking of the robots recently began to attract much attention in the robotics community. Teixid et al. estimated position of circles from a laser scan, stating that attaching a circular marker to a mobile platform facilitates its detection [1]. Yet, their sensor was static and external to the robot, and the experiments limited to a case when the robot has been moved manually, not reflecting its real, dynamic motion. Within the framework of distributed robotic systems, Huang et al. studied multi-robot cooperative localization with a extended Kalman filter (EKF) and two new observability constrained OC-EKF estimators [2]. In their experiments they synthetically produced the relative range and bearing measurements using the differences in the positions of the robots,

Fig. 1: Left: The Mbot robots. Right: Map of the environment with area allowed for navigation marked in red.

which were recorded by the overhead camera. He and Du tracked dynamic objects to anticipate possible collisions and choose obstacle avoidance policy, but did not distinguish between a robot or other moving obstacle, such as human [4].

Examples of relative positioning systems operating without wireless communication are also present in the literature. For instance, Fredslund and Mataric performed detection of a neighboring robot using combination of a laser and a camera. The range was acquired using a laser, while the bearing was obtained by reading the tilting of a panning camera and keeping in its center the image of a color-coded marker attached to the other robot. In the work by Soares et al. robots operated in an underwater medium, which makes it problematic to both communicate and localize, yet the local acoustic ranges provided enough information to serve as a basis to perform a formation maneuver [5]. Pugh et al. presented an onboard relative positioning module for miniature robots, which operated using modulated infrared signals [6]. The module enabled accurate multi-robot formations, but was tailored to a specific robotic platform.

In this paper we present a method that enables one robot navigating in an environment shared by other robots to obtain relative positions of the other formation members. Our work relies exclusively on information obtained by a robot from two on-board laser range finders without any assumptions neither about the motion of the robot itself or the other robots or the environment. Since it does not rely on wireless communication, it relaxes the dependence on network reliability to deliver positioning data. For example, if the robots can communicate their self-positioning data, but the communication network is unreliable or has restricted bandwidth, the robots may exchange the data only sporadically and complement the information with the relative positioning system. Combination of both systems would be highly advantageous in a structured indoor environment, resulting in more robust solution than any of the two used separately. Our method is applicable to a general case, such as navigating in an environment with dynamic obstacles that are not part of the multi-robot system but nonetheless may impair the performance. While our detection technique provides an accurate instantaneous estimation, stochastic dynamic filtering helps to deal with random noise, occlusions, and false positives. Problems associated

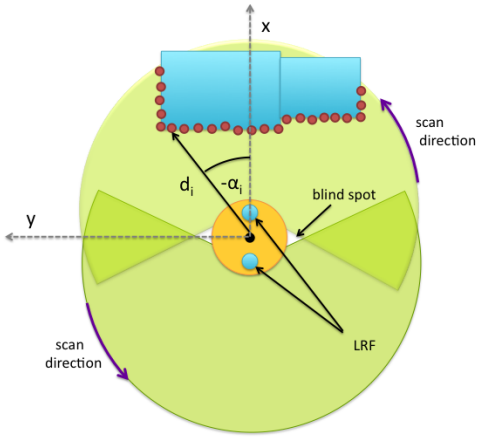


Fig. 2: Illustration of the sensor measurements. The x axis is aligned with the front of the robot (a yellow circle). The two LRF are located (27.5, 0) cm and (-27.5, 0) cm from the origin.

with relative-positioning systems, such as occlusions and limited field-of-view restrain their scalability, however the methods presented in this work could serve as additional back-up positioning system in case of communication issues or even as a fundamental information source, complemented by the data from a more reliable but computationally expensive or network-heavy nodes.

This paper is organized as follows. In Section 2 we present an overview of our robotic platform and its sensing capabilities. In Section 3, we describe the robot detection method using two-dimensional laser measurements and cover in detail the robot tracking algorithm. Experiments with real robots are presented in Section 4. We draw conclusions in Section 5.

## 2 System Architecture

Our method has been devised and tested on the MBot robots (Fig. 1) designed within the frame of the ongoing FP7 European project MOnarCH (Multi-Robot Cognitive Systems Operating in Hospitals) with the goal of introducing social robots in real human environments and studying the establishment of relationships between them[3].

The robot is equipped with navigation, perception, interaction and low-level safety sensors. For navigation and particularly for mapping, localization and obstacle avoidance, the robot fuses measurements provided by laser range finders, odometry encoders and IMU sensors. The methods presented in this paper are based on the readings of two laser range finders (LRF) URG-04LX-UG01 manufactured by Hokuyo. Each of the two-dimensional LRFs measures 683 distance points in a range from -120° to 120°, where 0° corresponds to the front of the sensor [7]. The sensors are mounted inside the robot at a height of approximately 13 *cm* above the ground, one in front of the robot heading towards the front and one on the back heading backwards (see Fig. 2). Altogether, both LRFs provide 4 *m* sensing distance and 360° field-of-view. After translation

[3] MOnarCH, FP7, FP7-ICT-2011-9-601033 (http://monarch-fp7.eu)

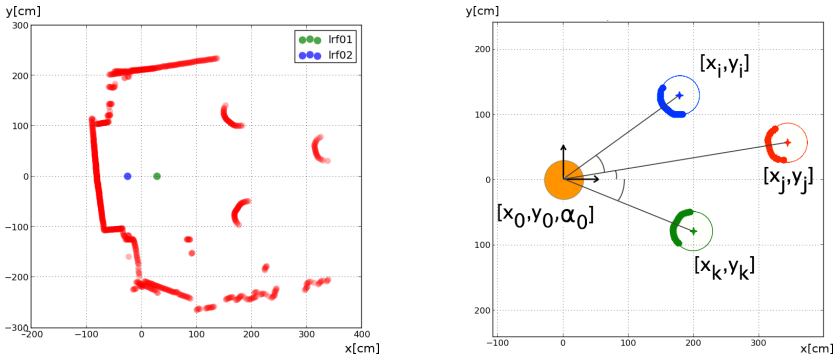


Fig. 3: Left: Raw laser readings in the robots coordinate system. The blue and the green dots indicate positions of the two laser range finders. Right: Result of the circle fitting algorithm. The orange circle is the detecting robot.

to the robot's central coordinate system, each individual distance point of the LRF reading is represented in the polar coordinates $S = (d_i, \alpha_i)$, where $d_i$ is the distance of the data point from the origin of the robots coordinate system and $\alpha_i$ is the relative angle. Example of a raw laser scan can be seen in Fig. 3 (left).

The navigation of the MBot robots is based on a standard occupancy grid map, serving for both motion planing [8] and self-localization [9], the latter obtained by combining odometry with AMCL[4]. The robot moves using mecanum wheels, an omnidirectional locomotion system with a maximum speed of 2.5 $m/s$ and maximum acceleration of 1 $m/s^2$. A complete description of the MBot robot can be found in [9].

## 3 Estimation and Tracking Method

We propose a methodology to detect a circular marker on a two-dimensional plane using two laser range finders mounted on a mobile platform. We use one MBot robot to locate all the others in the range of the LRF, as shown in Fig. 3. Although in our case we validated our approach using up to three robots, there are no intrinsic limitations of the method in terms of number of robots, as long as the other platforms are clearly distinguishable and not occluded. The shape of the robot base (approximated by a circle), known a priori, serves as a model for the detection algorithm. Note however that this is not a limitation of the algorithm in terms of generalization as such geometric assumptions can be easily customized for any other robotic platform. In this section, we describe the steps required to process the raw laser scan data to estimate the position of the observed robot. We focus on a scenario involving only two robots and indicate where the algorithm branches for generalization to a multi-robot case. An outline of the estimation technique is presented in Algorithm 1. We denote $S$ the set of Lidar points, $CL$ the set of point-cloud clusters of a size $n_{CL}$ and $n_c$ number of circles fitted to the data.

### 3.1 Data Clustering and Selection

As described in Section 2, the laser scan after initial pre-processing is stored as a tuple $(d_i, \alpha_i)$ for each data point $i$. Each scan sequence produces a map of the

[4] AMCL, (http://wiki.ros.org/amcl, retrieved 16 June 2015).

immediate neighborhood with distinguishable shapes, such as walls or objects cluttered within the sensing range (see Fig. 3). After deletion of isolated points caused by a mixed pixel phenomenon [10], which generates a measured range resulting from a combination of the foreground and the background objects, we cluster the scan points using simple nearest neighbor classification by moving a sliding window of a 3-point size and discerning the separated objects using a minimal Euclidean distance threshold. The thresholds are highly implementation dependent, in our case are $T_{sec} = 20, T_{sel}^{max} = 200$ and $T_{sel}^{min} = 20$. The acquired segments allow the algorithm to clearly distinguish among different objects, which might be directly adjacent to the walls. Being only attentive to the circular candidates of a certain size, the large objects characterized by excessive segment dimensions as well as object without curvature (the walls) are discarded.

**Algorithm 1. estimation_and_tracking($\boldsymbol{d}, \boldsymbol{\alpha}$)**

$S = (\boldsymbol{d}, \boldsymbol{\alpha})$
$S \leftarrow delete\_outliers(S)$
$n_{CL} = 0$
$CL[n_{CL}] \leftarrow (d_0, \alpha_0)$
for $k = 2$ to $len(\boldsymbol{d})$: # Point cloud clustering
  if $dist(s_{k-2}, s_{k-1}) > T_{sec}$ and $dist(s_{k-1}, s_k) > T_{sec}$
    $CL[n{+}{+}] \leftarrow S_k$
  else
    $CL[n] \leftarrow S_k$
for $k = 0$ to $n_{CL}$: # Cluster selection
  if $dist(CL[k], CL[len(CL)]) > T_{sel}^{max}$
      or $dist(CL[k], CL[len(CL)]) < T_{sel}^{min}$
    $CL \leftarrow delete(CL[k]), n_{CL} - -$
$n_c = 0, [\boldsymbol{x}_c, \boldsymbol{y}_c] = \emptyset$
for $k = 0$ to $n_{CL}$: # Circle fitting
  $[\boldsymbol{x}_c, \boldsymbol{y}_c] \leftarrow fit\_circle(CL[k]), n_c{+}{+}$
$[\boldsymbol{x}_c, \boldsymbol{y}_c] \leftarrow merge\_overlapping\_circles([\boldsymbol{x}_c, \boldsymbol{y}_c], n_c)$
$[\boldsymbol{x}_c, \boldsymbol{y}_c] \leftarrow is\_in\_area\_allowed([\boldsymbol{x}_c, \boldsymbol{y}_c], n_c)$
$[\boldsymbol{x}_c^g, \boldsymbol{y}_c^g], n_c \leftarrow global\_to\_local\_coord([\boldsymbol{x}_c, \boldsymbol{y}_c])$
$[\boldsymbol{r}, \boldsymbol{\phi}] \leftarrow euclidean\_to\_polar[\boldsymbol{x}_c^g, \boldsymbol{y}_c^g]$

### 3.2 Circle Fitting

The pre-selected clustered point clouds, having approximately the size of the object to be detected, are each in turn supplied into a least squares optimization algorithm extended with a modification of the Levenberg-Marquardt algorithm [11]. We minimize an objective function $F$ over a space of only two parameters, namely the coordinates $(x_c, y_c)$ of the center of the circle, expressed as a relative polar coordinate in respect to the center of the detecting robot:

$$F(\mathbf{d}, \boldsymbol{\alpha}) = \sum_{i=1}^{N^K} \left( \sqrt{(\varepsilon_x d_i cos(\alpha_i) - x_c)^2 + (\varepsilon_y d_i sin(\alpha_i) - y_c)^2} - R \right)^2 \quad (1)$$

where $N^K$ is the number of data points present in the current cluster $K$, $R$ is the radius of the circular object (in this case the robot radius) and $\varepsilon_x$ and $\varepsilon_y$ are the signs reflecting the scan angle convention of the frontal or the rear laser. We use the size of the robot base $R = 60\ cm$. The result of the circle fitting algorithm shown in Fig. 3 illustrates the situation from the Fig. 1 (right), where the robot on the back detects the three robots present in the experimental area.

### 3.3 Candidate Target Validation

A structured indoor environment, such as the area where we conduct our experiments (see Fig. 1), may generate false detections. Building features as support columns or trash bins unnecessarily increase the number of false positives, therefore we take advantage of the fact that the environment is known a priori. The robots of the MBot size can physically access only limited section of available space; for instance, they cannot move under the structural elements or inside the narrow spaces. These features generate most of the false detections, so we reduce the search space to areas accessible by the MBot robots (Fig. 1 (right)).

Fig. 1 (right) shows a fragment of a standard occupancy grid map available for the robot for the purpose of self-localization, based on which at any point in time, it can estimate its own pose. Having the circle fitting algorithm return a set of coordinates $(x_c, y_c)$ of the detected robots in the local coordinate frame of the detecting robot, we can find a precise location of the robots on the map:

$$\begin{bmatrix} x_c^g \\ y_c^g \end{bmatrix} = \begin{bmatrix} cos(\alpha_0) & -sin(\alpha_0)) \\ sin(\alpha_0) & cos(\alpha_0) \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} x_0^g \\ y_0^g \end{bmatrix} \tag{2}$$

where the superscript $g$ refers to the global coordinate frame and the $[x_0, y_0, \alpha_0]$ is the pose of the observing robot in the global frame. Along the presented methodology we developed a simple tool which allows to analyze a previously marked occupancy grid map to determine whether a given $[x, y]$ point on the map is marked as *accessible*. The map can be edited using any of the image manipulation programs (Fig. 1 shows an example of such a map). We assume that if the $(x_c, y_c)$ is located within the marked area, it becomes a valid measurement of the position of the robot $R_k$, i.e. $(x_k, y_k)$.

### 3.4 Range and Bearing

The detection, at this point assumed being a valid relative position of another robot, necessitates an estimation of the associated positioning error. In this section, we present such procedure resulting in the evaluation of the error associated with the detection of the circle position, to be further used in a Bayesian estimator for tracking purposes.

The range and bearing measurement error from the robot $R_i$ to $R_j$ is defined as a difference between the actual relative positions and the estimated values:

$$\tilde{r}(\hat{r}_{ij}, p_i, p_j) = \hat{r}_{ij} - r_{ij}(p_i, p_j) \tag{3}$$

$$\tilde{\phi}(\hat{\phi}_{ij}, p_i, p_j) = \hat{\phi}_{ij} - \phi_{ij}(p_i, p_j) \tag{4}$$

where $p = [x, y]$ is the Euclidean representation of the position and the range and bearing are a transformation of the Euclidean representation in polar coordinates:

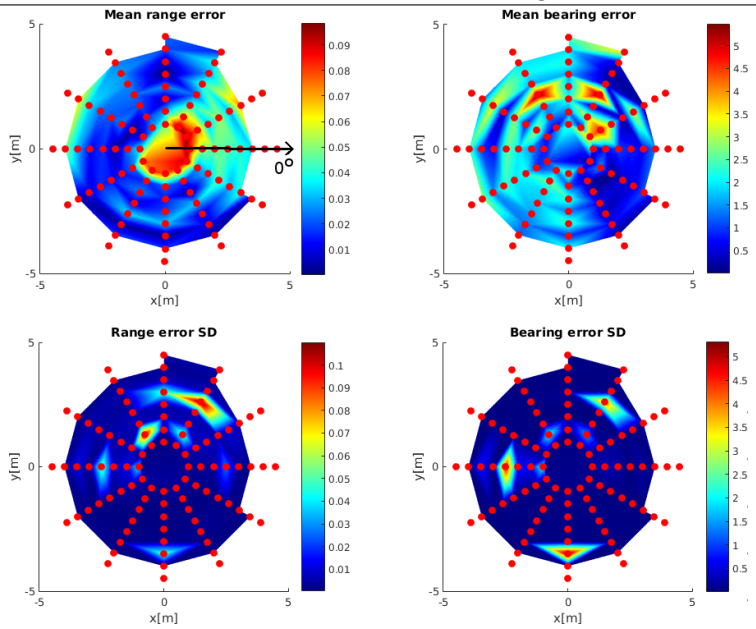


Fig. 4: Error map showing the accuracy (top) and the precision (bottom) of the relative positioning method. The scale for the range error is in meter; the scale for the bearing error is in degree. The heading of the robot is indicated with a zero degree angle.

$$\begin{bmatrix} r_{ij} \\ \phi_{ij} \end{bmatrix} = T_e^p(p_i, p_j) = \begin{bmatrix} r(p_i, p_j) \\ \phi(p_i, p_j) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\ atan2((y_i - y_j), (x_i - x_j)) \end{bmatrix} \quad (5)$$

The relative positioning error has been evaluated with a set of systematic experiments, where one robot to be detected has been placed manually at distances from 1.5 m to 5 m in steps of 0.5 m and at angles from 0° to 360° in steps of 30°. A total of 100 scans has been acquired for each location. A summary of the results is presented in Fig. 4 as two-dimensional linearly interpolated color maps. In general, we note that the range measurement error is smaller on the sides of the observing robot (i.e. for 60°-120°and 240°-300°), where the observed robot is within the sensing range of both laser range finders. The bearing error has a tendency to increase with the distance. It can be noticed that the frontal sensor has a reduced sensing range of 4 m. Composition of its measurements with those of the rear laser extends the sensing range to 4.5 m on the sides, the same as it is for the rear laser only. The color map distribution serves in the Kalman filter tracking for the estimation of the variance associated with the positioning error measurements. Thus, for the range and bearing measurements we assume the observation noise to be represented by a Gaussian probability density function. We sample the mean $\mu$ and the variance $\sigma$ from the polar error (Fig. 4).

### 3.5 Multi-Target Tracking

The tracking, depending on an application, can be performed in the global coordinate frame or locally, relatively to the tracking robot. The latter approach is typically implemented for tasks requiring the robots to know only the relative positions of the teammates, such as formation control [12]. We perform the

tracking in the global coordinate frame because of its generality, that allows the robots to fuse the information coming from various sources.

For our Kalman estimator, we assume a constant velocity and randomized Gaussian acceleration motion model. The state equations of the moving robot are:

$$\begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} + \varrho(k), \qquad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \epsilon(k) \tag{6}$$

The state vector of the observed robot is $x = [x, y, \dot{x}, \dot{y}]^T$, the observation model is $z = [x, y]^T$, $\varrho(k) \sim \mathcal{N}(0, Q(k))$ is a process noise assumed constant with $Q = [0.1\ 0.1\ 0.1\ 0.1]^T$ and $\epsilon(k) \sim \mathcal{N}(0, T_p^e(\xi))$ is a measurement noise with zero mean and the variance sampled from the error map in Fiq. 4 transformed into the Euclidean space.

## 4 Experiments

We considered four distinct scenarios to determine the performance of the detection and tracking method presented above. They demonstrate the cases of detection with one moving robot and one static detecting robot (I), one moving robot and one detecting robot following it (II), fusion of measurements of one observed robot from two detecting robots (III) and detection of two robots by one static robot (IV). Experiments have been carried out in a lab setting cluttered with appliances. For each scenario, we performed 12 runs (unless stated otherwise) and present the time-wise average as well as an aggregate of the results. The experiments took place in an area frequently visited by humans, who either traversed the space or walked around the room performing their duties and not being attentive to the motion of the robots. While the presence of the people caused instantaneous false detections, the Kalman filter was able to reduce their impact to a minimum. We obtain our ground-truth from the self-localization system provided by the navigation software (see Section 2 and [9]). The self-positioning capabilities of the robots, characterized by an accuracy of about 10 cm, serves as a ground truth measurement used in the performance evaluation.

The accuracy of the relative positioning system has been evaluated using standard root-mean-square error between the estimates and the actual values:

$$E_{AV}(t) = \sqrt{\frac{1}{n_d} \sum_{i=1}^{n_d} \|p_i(t) - \hat{p}_i(t)\|^2}\ , \quad E_T = \frac{1}{T_{max}} \sum_{j=1}^{T_{max}} (E_{AV}(j)) \tag{7}$$

The $E_{AV}(t)$ is the time-wise average error over the number of experiments $n_d$ and the $E_T$ is the total average error over the duration of a single run.

### 4.1 Results

***I. Single robot detection from a static platform.*** In the first scenario we fix the position of the detecting robot and move the observed robot on a simple trajectory as shown in Fig. 5 (left). The sampling frequency is fixed to 1 Hz to reflect the effect of the Kalman filter. The tracking estimates of the observed

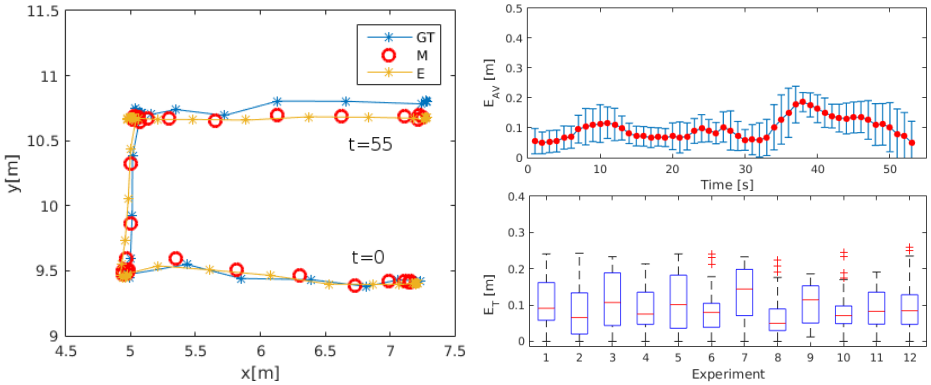


Fig. 5: Single robot detection from a static platform. (Left) Sample trajectory of the observed robot and the position estimate. The observing robot position is $[5.9, 7.1]$. $GT$ is the ground truth, $M$ are the measurements and $E$ the estimate. This convention remains valid throughout the paper. (Top right) Time-wise average and SD of the error. (Bottom right) The average error aggregated for all the runs.

robot follows very closely the position reported by its self-positioning system, with the error not exceeding 20 cm (see Fig. 5, top right). The Fig. 5 (bottom right) shows the aggregate error of each experiment. The total average error during the period when the robot was detected is around 10 cm, slightly higher than expected given the error map. We hypothesize that the increased error might have been caused by approximation of the robot base with a circle - the actual body of the robot is flattened on the sides.

***II. Single robot detection from a mobile platform.*** The second case studies the impact of the detecting platform movement on the tracking system. The observed robot follows the trajectory shown in 6 (left) and the detecting robot keeps a constant distance to the estimated position using a simple proportional feedback controller. The sampling frequency is 1 Hz. The detection error is bounded and level off (Fig. 6, bottom right); the average error $E_{AV}$ alternates around the value of 20 cm (Fig. 6, top right). The increase of the detection error in comparison with the static case can be caused by inaccuracies associated with the self-positioning system of the detecting robot. During two runs (outliers of 7 and 9 in Fig. 6 bottom right) we encountered false positive detections that proved necessity of a Kalman filter to reduce their impact on a final performance.

***III. Detection and fusion from two static platforms.*** The following set of experiments is conducted as to approximate the effect of sensor fusion from two sensing platforms. To be precise, one observed robot moves around the arena and the two observing robots are static and positioned so that the observed robot for a certain amount of time is outside of the sensing range of at least one of them. The two detecting robots are communicating their Kalman filter estimates and each of them fuses the data for a better approximation of the position of the observed robot. For the two observing robots $R_i$ and $R_j$, estimating the position of the robot $R_k$ as $[x_i^k, y_i^k]$ and $[x_j^k, y_j^k]$ respectively, the result of the sensor fusion can be calculated by combination of the Gaussians:

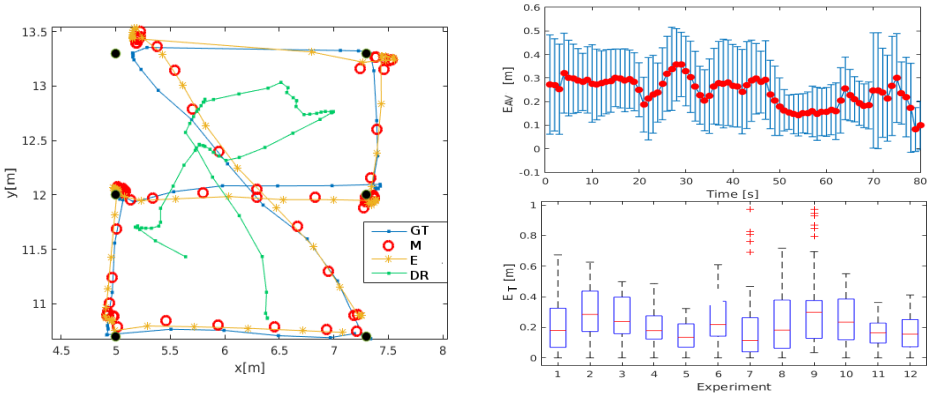


Fig. 6: Single robot detection from a mobile platform. (Left) The trajectory of the observed robot (True), the measurements (Meas.) and the estimate (Estim.), *DR* is the trajectory of the following (detecting) robot.

$$\frac{1}{\sigma_T^2} = \frac{1}{\sigma_{ik}^2} + \frac{1}{\sigma_{jk}^2} \ , \qquad x_T^k = \sigma_T^2 \left(\frac{x_i^k}{\sigma_{ik}^2} + \frac{x_j^k}{\sigma_{jk}^2}\right) \tag{8}$$

where $\sigma_{mn}^2(r_{mn}, \phi_{mn})$ is the standard deviation associated with the measurement of the robot $R_m$ to the robot $R_n$ obtained from the error map (Fig. 4). The method is fully scalable, up to the capacity of the communication network and can easily accommodate additional nodes. An additional advantage of this method emerges when the observed robot ventures outside of the range of one of the detecting robot. For example, if robot $R_i$ does not obtain an observation, its variance is set to infinity $\lim_{\sigma_{ik}^2 \to \infty} \sigma_T^2 = \sigma_{jk}^2$, the corresponding estimate levels out and the fused estimate in the two-node case attains value of the other measurement as in $\lim_{\sigma_{ik}^2 \to \infty} x_T^k = x_j^k$.

Fig. 7 (left) shows the estimates recorded by the two detecting robots and the fused estimate. The detection range of robot $R_1$ covers only the upper part of the area (above 10.6 m), below which the fused estimate follows directly that provided exclusively by robot $R_2$. Otherwise, the fusion significantly improves the overall estimate, lowering the error and reducing its variance (Fig. 7, top right). The difference in comparison to single-platform case is clearly visible in Fig. 7 (bottom right), where the largest error during the runs did not exceed half of that achieved using a single observing robot.

***IV. Multi-robot detection from a static platform*** The final experiment focuses on a multi-robot detection scenario. We perform 10 runs of an experiment with 2 robots moving on a rectangular trajectory (see Fig. 8, left), with the detecting robot placed in the middle of the arena. Although the tracking performs very well on average (Fig. 8, right), there is a number of false positives (red outliers), caused principally by random objects appearing as having a circular surface from a specific observation point. While these outliers are usually eliminated by the Kalman filter, in this case they might have been caused by temporary occlusions or faulty data association. Occasional false positives do not impair the tracking, but their cause needs to be further investigated.

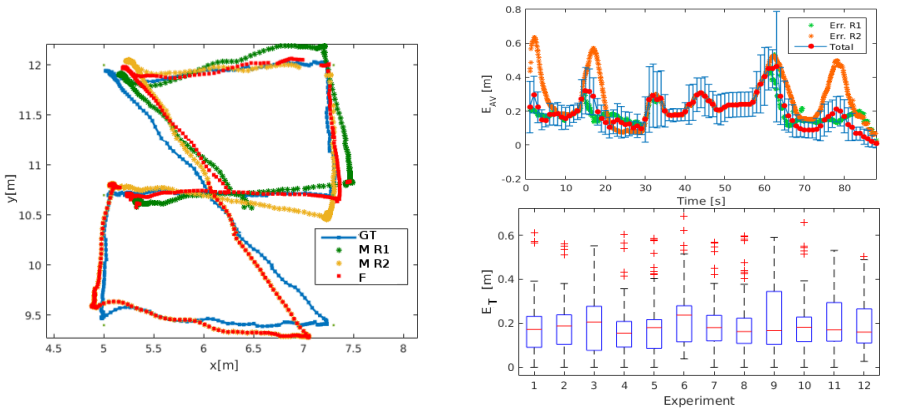


Fig. 7: Detection and fusion from two static platforms. The detecting robots $R_1$ and $R_2$ are positioned at $(6.0, 6.7)$ and $(6.4, 13.5)$ respectively. $F$ is the fused estimate. (Top right) shows the average errors of both robots over the runs.

## 5 Conclusions

In this work we presented a lidar-based relative position estimation and tracking. A distinguishing feature of our work is the fact that each robot only relies on two-dimensional scan provided by a laser range finder for sensory information. By exploiting simple geometric features of the individual robotic platform, we were able to reliably estimate and track the position of the other robots present in the environment. Our method can be easily extended for tracking objects of various sizes and shapes by changing model of the object, allowing for detection and tracking of heterogeneous robots. We evaluated our approach during systematic real-world experiments, where we studied the performance in scenarios involving various combinations of static and mobile robots.

Experimental results show that while the results in terms of accuracy are useful in the targeted range of 20-30 cm, further effort is needed to increase the reliability of the lidar-based relative positioning method, possibly improving on its individual components (classification, tracking etc.). We will further focus on increasing robustness of our method by adding a data association module for tracking multiple robots in cluttered arenas and be able to deal with moving obstacles that can be misclassified as robots. We intend to apply our work as a complementary tool in multi-robot behaviors. In particular, we plan to use the methods presented in this paper as an additional source of information in multi-robot formations behavior [12], significantly reducing its dependence on communication. In contrast to the systems that depend on the local communication, the relative positioning method suffers from the line-of-sight limitation, therefore merging the complementary strengths of both techniques is highly beneficial.

**Acknowledgements.** Supported by ISR/LARSyS Strategic Funds from FCT project FCT[UID/EEA/5009/2013] and FCT/11145/12/12/2014/S and FCT/PD /BD/105784/2014 and by European MOnarCH project FP7-ICT-9-2011-601033.

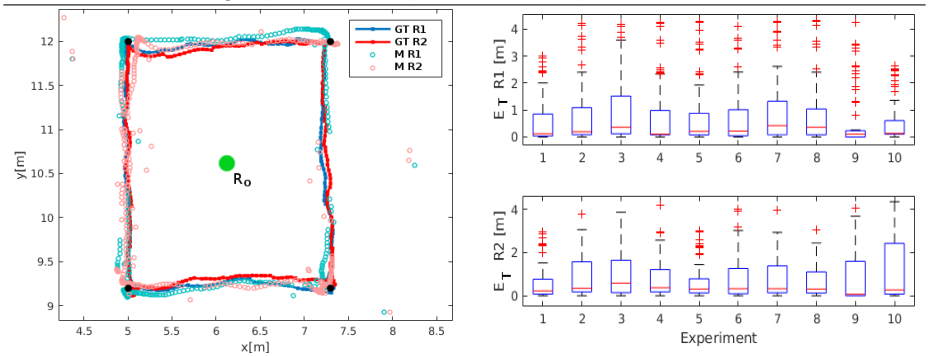


Fig. 8: Multi-robot detection from a static platform. The green mark shows the position of the observing robot at $[6.1, 10.7]$. (Right) The red outliers are the false detections.

## References


1. Teixid, M., Pallej, T., Font, D., Tresanchez, M., Moreno, J. and Palacn, J. *Two-Dimensional Radial Laser Scanning for Circular Marker Detection and External Mobile Robot Tracking*. Sensors 12, 16482-16497, 2012.
2. Huang, G. P., Trawny, N., Mourikis, A. I. and Roumeliotis, S. I., *Observability-based consistent EKF estimators for multi-robot cooperative localization*. Autonomous Robots, 30(1), 99-122, 2011.
3. Fredslund J. and Mataric M. J., *A general, local algorithm for robot formations*, IEEE Trans. on Robotics and Automation, special issue on Advances in Multi-Robot Systems, 18(5), 837-846, 2002.
4. He, F., Du, Z., Liu, X., and Ta, Y., *Laser Range Finder Based Moving Object Tracking and Avoidance in Dynamic Environment*. Proc. of IEEE Int. Conf. on Information and Automation, 2357-2362, 2010.
5. Soares J. M., Aguiar A. P., Pascoal A. M., and Martinoli A., *Joint ASV/AUV Range-Based Formation Control: Theory and Experimental Results*, Proc. of the 2013 IEEE Int. Conf. on Robotics and Automation, 5579-5585, 2013.
6. Pugh J., Raemy X., Favre C., Falconi R. and Martinoli A., *A Fast Onboard Relative Positioning Module for Multirobot Systems*. IEEE/ASME Trans. on Mechatronics, 14(2), 151-162, 2009.
7. *Scanning Laser Range Finder URG-04LX-UG01 Specifications*. Available online: https://www.hokuyo-aut.jp/02sensor/07scanner/download/pdf/URG-04LX_UG01_spec_en.pdf. Accessed on 22 May 2015.
8. Ventura R. and Ahmad A., *Towards optimal robot navigation in urban homes*, Proc. of the 18th RoboCup Int. Symposium, 2014.
9. Messias J., Ventura R., Lima P., Sequeira J., Alvito P., Marques C. and Carrico P., *A robotic platform for edutainment activities in a pediatric hospital*, Proc. of the 2014 IEEE Int. Conf. on Auton. Robot Sys. and Competitions, 193-198, 2014.
10. Okubo Y., Ye C., and Borenstein J., *Characterization of the Hokuyo URG-04LX laser rangefinder for mobile robot obstacle negotiation*. SPIE Def., Sec., and Sens.. Int. Soc. for Opt. and Phot., 2009.
11. More, J.J., *The Levenberg-Marquardt algorithm: Implementation and theory*. Numerical analysis. Springer Berlin Heidelberg, 105-116, 1978.
12. Das, A. K., Fierro, R., Kumar, V., Ostrowski, J. P., Spletzer, J., and Taylor, C. J., *A vision-based formation control framework*. IEEE Transactions on Robotics and Automation, 18(5), 813-825, 2002.