Robust Range-Only Mapping via Sum-of-Squares Polynomials

Lionel Génevé, Adlane Habed, Edouard Laroche, Olivier Kermorgant

Robust Range-Only Mapping via Sum-of-Squares Polynomials

Lionel Génevé * **Adlane Habed** * **Édouard Laroche** *
Olivier Kermorgant **

* *ICube Laboratory, University of Strasbourg-CNRS, Strasbourg, France (e-mails: {l.geneve, laroche, habed}@unistra.fr)*
** *LS2N, École Centrale de Nantes, Nantes, France (e-mail: olivier.kermorgant@ec-nantes.fr)*

Abstract: This work presents a new approach for mapping static beacons given only range measurements. An original formulation using sum-of-squares and linear matrix inequalities is derived to test if a measurement is inconsistent with a bounding box containing the beacon position. By performing this test for each range measurement, it is possible to recursively eliminate incompatible boxes and find the smallest consistent box. The box search is done with a breadth-first search algorithm that recursively prunes inconsistent boxes and splits the others to narrow the estimation. The validity of the method is asserted via simulations and compared to other standard mapping methods. Different levels and types of noise are added to evaluate the performances of the algorithm. It resulted that the approach accommodates very well classical zero-mean white Gaussian noises by adaptating the ratio of tolerated outliers for the consistency check, but fails to handle additive biases.

Keywords: Mapping, Range-Only, Beacon, LMI, SoS

1. INTRODUCTION

In robotics, the mapping problem consists in constructing a representation of the environment. The latter can then be used by a robot to plan its trajectory, navigate and avoid obstacles. When there is no a priori information on the robot positions, this is referred to as the simultaneous localization and mapping (SLAM). This topic has already been extensively covered in the literature even for the less common range-only (RO) case, where range measurements are the only exteroceptive information available, as it is the case in this work. For instance, Djugash and Singh (2009) use an extended Kalman filter (EKF) with a polar representation of the beacon positions. Later on, Caballero et al. (2010) improved the formulation with a mixture of Gaussians to obtain an undelayed initialization of the beacons in the filter. The formulation was then extended to the 3D case with a spherical representation (Fabresse et al., 2013). Blanco et al. (2008) follow the same idea but using a particle filter framework instead. In Boots and Gordon (2012), a spectral learning approach is proposed to solve the problem. More related to our work, Di Marco et al. (2004) use a set theoretic approach and unknown-but-bounded errors to recursively estimate guaranteed uncertain regions of the robot and beacon positions. In the same context, Jaulin (2011, 2009) uses interval analysis and constraint satisfaction methods to propagate and reduce the intervals representing the estimated variables.

In this work, an interval representation of the estimated variables is also adopted. However, contrary to the previously cited works, here only the pure mapping problem is considered. This means that we make the assumptions that the robot positions are known without errors. However, our approach could also be used for localization and possibly for RO-SLAM problem.

Regarding the mapping problem, a popular technique introduced by Moravec and Elfes (1985) is the occupancy grid method. It uses a probabilistic framework to update the cell occupancy probability, and provides a 2D or 3D discretized representation of the environment. In order to localize an underwater vehicle, Olson et al. (2006) use a grid representation and a voting scheme strategy to obtain a first estimate of the beacon positions. By computing the intersection of two range measurements, the two solutions obtained vote for a cell in the grid. Ultimately, the cell with the largest support is considered as the estimated beacon positions. This estimate is then used as an initialization in an EKF localization algorithm. Another technique that can be used for mapping is the trilateration algorithm (Zhou, 2009). The principle is to combine at least three range measurements to form and solve a least-squares problem. The solution can then be refined by any nonlinear optimization algorithm. The main drawback of most of the presented methods is their lack of robustness and sensitivity to outliers. As we will see later, our method can nicely accomodate the outliers by adjusting a parameter. Compared to most of the previous works, the proposed method works currently offline by first collecting all the range measurements along the robot trajectory and then processing them. But this allows to adjust the number of tolerated outliers in the search process and to refine the estimated beacon positions. Thus, the proposed method is intended for retrieving the position of indoor or outdoor beacons in a garantueed way by returning an interval enclosing the true position of the beacons when the measurements are corrupted by many outliers.

In this work, the mapping problem with known robot positions is treated with an original formulation combining linear matrix inequalities (LMI) (Nesterov and Nemirovski, 1994) and sum-of-squares (SoS) polynomials (M.D. Choi, 1995; Lasserre, 2007). LMI (Gahinet et al., 1995) are extensively used in control engineering and became a powerful tool in many domains. Moreover, thanks to the works of Nesterov and Nemirovski (1994) on the interior point method, LMI can now be efficiently solved. The method presented in this paper is inspired by the work developed by Pani Paudel et al. (2015), initially desgined for point-to-plane registration, and adapted here for the mapping problem. Using only range measurements between a mobile robot with known positions and several unknown static beacons scattered in the environment, the objective is to recover the location of the beacons. This is done by checking that all the corresponding range measurements are compatible with a bounding box supposed to contain the beacon position. The consistency test is done by transforming the range measurement constraints into a polynomial inequality. If this polynomial is SoS, the corresponding measurement is inconsistent. Checking that a polynomial is SoS is then efficiently solved as a LMI feasability problem. Notice that this technique does not require a noise model or hypotheses for the measurements, and that it is possible to make it robust with respect to erroneous measurements occuring in real cases by controlling the percentage of measurements that do not satisfy the consistency test.

The rest of this paper is organized as follows. Section II introduces the problem and the formulation used throughout the paper. Section III describes the algorithm and explains in detail the consistency check procedure. Section IV presents the various simulation results obtained with increasing noise levels. Finally Section V concludes the paper and gives some future perspectives.

2. PRELIMINARIES

This section briefly introduces the concepts used by the proposed approach and then describes how they are integrated in the formulation.

2.1 Technical background

Set-membership functions: For a given interval $[\underline{x}, \overline{x}]$, let us consider the membership function $g(x)$ such that:

$$g(x) = (x - \underline{x})(\overline{x} - x) \tag{1}$$

The function is positive inside the interval and negative outside. Thus, $x \in [\underline{x}, \overline{x}]$ iff $g(x) \geq 0$.

A 2D box is defined as the product of two intervals:

$$\mathbb{B} = \left\{ \mathbf{x} = (x, y) \in \mathbb{R}^2 \,\middle|\, (x, y) \in [\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}] \right\} \tag{2}$$

Sum-of-Squares (SoS) and Linear matrix inequalities (LMI): A scalar polynomial function $p(x)$ is SoS if there exists a polynomial $q(x)$ such that:

$$p(x) = \sum_i q_i(x) q_i(x)$$

A polynomial function $p(x)$ is positive if it is SoS. This notation can be easily extended to the case of matrix inequalities. Moreover, if we have:

$$\mathbf{P}(\mathbf{x}, \mathbf{l}) = \mathbf{Q}^T(\mathbf{x})\mathbf{M}(\mathbf{l})\mathbf{Q}(\mathbf{x})$$

with $\mathbf{M}(\mathbf{l})$ a symmetric matrix function of $\mathbf{l}$ then $\mathbf{P}(\mathbf{x}, \mathbf{l})$ is positive if $\mathbf{M}(\mathbf{l}) \geq 0$ [1] and $\mathbf{P}(\mathbf{x}, \mathbf{l})$ is said to be SoS. In the case where $\mathbf{M}(\mathbf{l})$ is affine in $\mathbf{l}$, $\mathbf{M}(\mathbf{l}) \geq 0$ is called a linear matrix inequality. Checking the positivity of a polynomial function $\mathbf{P}(\mathbf{x}, \mathbf{l})$ can thus be solved via a LMI. Finally, the positivity can be restricted to a domain. For instance in 2D: $\mathbf{P}(\mathbf{x})$ is positive over $\mathbb{B}$ if there exists positive scalars σ_x and σ_y such that: $\mathbf{P}(\mathbf{x}) - \sigma_x g_x(\mathbf{x}) - \sigma_y g_y(\mathbf{x})$ is SoS, where $g_x(\mathbf{x})$ and $g_y(\mathbf{x})$ are the membership functions for the x and y components of $\mathbf{x}$.

2.2 Application to the approach

For the mapping problem, the trajectory of the robot is known and represented by a set of K robot positions:

$$\mathbf{x}_{r_k} = (x_{r_k}, y_{r_k})^T, \; k \in [0, K]$$

Note that the orientation information is not used here. The objective is to find the position of the N static beacons defined by:

$$\mathbf{x}_{b_i} = (x_{b_i}, y_{b_i})^T, \; i \in [1, N]$$

The proposed method is based on the ability to determine if a given box does not enclose the true position of the beacon. Thus, using the notation of (1) and (2), a box is represented by:

$$\mathbb{B}_i := \left\{ \mathbf{x}_{b_i} \in \mathbb{R}^2 \,\middle|\, g_x(\mathbf{x}_{b_i}) \geq 0, \; g_y(\mathbf{x}_{b_i}) \geq 0 \right\}$$

with:

$$\begin{cases} g_x(\mathbf{x}_{b_i}) = (x_{b_i} - \underline{x_{b_i}})(\overline{x_{b_i}} - x_{b_i}) \\ g_y(\mathbf{x}_{b_i}) = (y_{b_i} - \underline{y_{b_i}})(\overline{y_{b_i}} - y_{b_i}) \end{cases}$$

The range measurements between the robot and the beacons are the only information available to recover the beacon positions $\mathbf{x}_{b_i}$. We define the constraint $f_{k,i}(\mathbf{x}_{b_i})$ as the difference between the squared predicted range $d_{k,i}^2$ and the squared measured range $r_{k,i}^2$ between the robot pose at time k and the beacon i:

$$f_{k,i}(\mathbf{x}_{b_i}) = r_{k,i}^2 - d_{k,i}^2 \tag{3}$$

with $d_{k,i}^2 = (x_{b_i} - x_{r_k})^2 + (y_{b_i} - y_{r_k})^2$. A measurement is consistent if $f_{k,i}$ crosses zero at some point inside the box representing the beacon position. It should be noticed that the constraint (3) is quadratic with respect to the unknown $\mathbf{x}_{b_i}$, which makes it interesting for a SoS approach. In the next section, this constraint will be exploited to develop a method retrieving a beacon box consistent with all or most of the range measurements, by testing its positivity over a given box.

3. ALGORITHM DESCRIPTION

In this section, we first give an outline of the algorithm. Then, we delve into the derivation of the LMI consistency test. Finally, the procedure to explore and search the smallest compatible box is presented.

[1] For a real symmetric matrix, $\mathbf{M} \geq 0$ means that all the eigenvalues of $\mathbf{M}$ are non-negatives and reals.

3.1 Algorithm outline

The proposed method attempts to find, for each beacon, the smallest box where all the range measurements (or a predefined percentage of them) respect the condition:

$$f_{k,i}(\mathbf{x}_{b_i}) = r_{k,i}^2 - d_{k,i}^2 = 0 \quad (4)$$

More precisely, we count the number of range measurements satisfying the condition (4) for some point inside that box (notice that these points may not be the same for all range measurements due to noise). The condition is verified by testing the positivity of the constraint (3) and by solving a LMI feasibility problem. If it is feasible, then the constraint (3) is SoS and the condition (4) is not respected. Thus the range measurement is an outlier. By controlling the number of outliers, it is possible to discard or accept the considered box. The accepted boxes are then split and the test is repeated for each of the new boxes. This process is iterated until the algorithm has discarded all the potential boxes and a smaller consistent box cannot be found. The consistency test performed for a given box and a range measurement is now detailed.

3.2 LMI consistency check

We define the polynomial function $h_{k,i}(\mathbf{x}_{b_i})$ constructed from the constraint (3) and the membership quadratic polynomial (1):

$$h_{k,i}(\mathbf{x}_{b_i}) = \lambda f_{k,i}(\mathbf{x}_{b_i}) - \sum_{j=\{x,y\}} g_j(\mathbf{x}_{b_i})\sigma_j \quad (5)$$

where $\lambda \in \mathbb{R}$ and $\sigma_j \in \mathbb{R}^+$. The variable λ is added to take into account the cases: $\pm f_{k,i} > 0$. The equation (5) can be rewritten in a quadratic form as:

$$h_{k,i}(\mathbf{x}_{b_i}) = \mathbf{Q}^T(\mathbf{x}_{b_i})\mathbf{G}(\lambda, \sigma_x, \sigma_y)\mathbf{Q}(\mathbf{x}_{b_i}) \quad (6)$$

where $\mathbf{G}(\lambda, \sigma_x, \sigma_y)$ is the Gram matrix (Powers and Wormann, 1998) of $h_{k,i}(\mathbf{x}_{b_i})$ and $\mathbf{Q}^T(\mathbf{x}_{b_i}) = (x_{b_i}, y_{b_i}, 1)$.

The method stated in Pani Paudel et al. (2015) is adapted: for a given box $[\underline{\mathbf{x}_{b_i}}, \overline{\mathbf{x}_{b_i}}]$, if there exists a real scalar λ and positive scalars σ_j, $j = \{x, y\}$, such that the polynomial $h_{k,i}(\mathbf{x}_{b_i})$ is SoS, then $\lambda f_{k,i}(\mathbf{x}_{b_i}) > 0$ for every $\mathbf{x}_{b_i} \in [\underline{\mathbf{x}_{b_i}}, \overline{\mathbf{x}_{b_i}}]$. In this case, $f_{k,i}$ is either strictly positive, either strictly negative [2] over the whole domain and thus contains no point inside the domain for which $f_{k,i} = 0$. The range measurement $r_{k,i}$ is then guaranteed to be an outlier within the considered bounds. Otherwise, $r_{k,i}$ is a potential inlier. Verifying that $h_{k,i}(\mathbf{x}_{b_i})$ is SoS can be done by checking that its corresponding real symmetric Gram matrix from (6) is positive semi-definite: $\mathbf{G}(\lambda, \sigma_x, \sigma_y) \geq 0$ (Powers and Wormann, 1998; M.D. Choi, 1995). The latter condition can be verified by solving a LMI feasability problem for the variables λ and σ_j. Indeed, the matrix $\mathbf{G}(\lambda, \sigma_x, \sigma_y)$ can be decomposed as:

$$\mathbf{G}(\lambda, \sigma_x, \sigma_y) = \lambda \mathbf{G}_1 + \sigma_x \mathbf{G}_2 + \sigma_y \mathbf{G}_3 \quad (7)$$

with:

$$\mathbf{G}_1 = \begin{pmatrix} 1 & 0 & -x_{r_k} \\ 0 & 1 & -y_{r_k} \\ -x_{r_k} & -y_{r_k} & x_{r_k}^2 + y_{r_k}^2 - r_{k,i}^2 \end{pmatrix} \quad (8)$$

$$\mathbf{G}_2 = \begin{pmatrix} 1 & 0 & -\frac{1}{2}(\underline{x_{b_i}} + \overline{x_{b_i}}) \\ 0 & 0 & 0 \\ -\frac{1}{2}(\underline{x_{b_i}} + \overline{x_{b_i}}) & 0 & \underline{x_{b_i}}\,\overline{x_{b_i}} \end{pmatrix} \quad (9)$$

$$\mathbf{G}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -\frac{1}{2}(\underline{y_{b_i}} + \overline{y_{b_i}}) \\ 0 & -\frac{1}{2}(\underline{y_{b_i}} + \overline{y_{b_i}}) & \underline{y_{b_i}}\,\overline{y_{b_i}} \end{pmatrix} \quad (10)$$

To complete the LMI problem, the positivity constraints on the σ_j must be added: $\sigma_x > 0$ and $\sigma_y > 0$.

In summary, if the LMI problem is feasible, then we have: $\mathbf{G}(\lambda, \sigma_x, \sigma_y) \geq 0$, thus $h_{k,i}(\mathbf{x}_{b_i})$ is SoS, meaning that the range measurement is an outlier. Otherwise, if the LMI solver didn't find any solution, the range measurement is a potential inlier. Now that it is possible to test if a range measurement is consistent with a given box, it remains to define a procedure to combine all the constraints and search for the smallest compatible box. The next paragraph presents this procedure based on a breadth-first search (BFS) algorithm that was used to find the boxes.

3.3 Complete algorithm

A breadth-first search algorithm (Lee, 1961) was developed to find the smallest box consistent with all the range measurements with respect to one beacon. Thus, for N beacons, the search has to be executed N times. The algorithm takes as input the robot positions, the range measurements associated with the beacon, an initial box and the thresholds to control the percentage of outliers and stop the search when a predefined accuracy is obtained. Starting with a large initial box, the algorithm recursively divides the boxes until they are not consistent anymore with the measurements. The consistency test is carried out for each range measurement, and if an outlier is detected, or if the percentage of tolerated outliers reaches a predefined threshold, then the box is discarded. The ability to control the percentage of outliers is very important when we deal with real measurements corrupted by noise, as it will be shown in the experiments. To determine which box has to be explored, a list of consistent boxes is maintained. If a box passes the test, it is added to the list, otherwise it is discarded. The search stops when the list is empty or when a given accuracy is reached. The accuracy threshold is defined with a minimal box area set to 0.01 m^2 during the simulations. One critical issue encountered during the simulations was the process of splitting a box into smaller boxes. When the true beacon position is near one of the inside borders created during the division process, the consistency check can fail and stop the search prematurely, resulting in a raw evaluation of the beacon position. It is even more problematic when noise is present. In the proposed algorithm, this problem is handled by using a two pass division. The interval is first split in four, and if none of the sub-boxes passes the test, another division scheme is used. This way, we were able to increase the robustness of the algorithm, but at the

[2] In fact only if $\mathbf{x}_{b_i} \in]\underline{\mathbf{x}_{b_i}}, \overline{\mathbf{x}_{b_i}}[$

cost of longer search. Other strategies are possible, such as using jointly a local method, and will be considered in future work to improve the process. In the next section, the experiments conducted to validate the approach are presented.

4. SIMULATION RESULTS

In order to evaluate the proposed mapping algorithm, simulations of different kinds were conducted. Four different datasets, were employed. The first two (Scenarios 1 and 2) were created to test the method for simple cases and without noise in order to validate the approach. They contain respectively 15 and 160 range measurements. The third dataset comes from a ROS/Gazebo simulation (Scenario 3), whereas the fourth is a publicly available dataset (Scenario 4) presented by Djugash et al. (2009). These last two datasets were used to validate that the proposed algorithm can also perform well in more challenging conditions with different levels of noise. The ROS/Gazebo dataset contains 6654 range measurements and was used with and without noise. The noise was generated following a zero-mean white Gaussian distribution with a standard deviation of 0.1 m. The scenario 4 dataset contains 3434 range measurements, and was used in three different configurations: i/ without noise, ii/ with noise and no bias, and iii/ with the original range measurements from the dataset. The values for the bias are between -2.6 and -2.9 m, whereas the standard deviations are between 1 and 1.2 m. The different noises are referred in the rest of this section as: Noise 1 when there is only a white Gaussian noise, and Noise 2, when there is a white Gaussian noise and additional bias. In order to evaluate the noise accomodation, different percentage (0%, 5%, 10% and 20%) of tolerated outliers were tested during the simulations. The algorithm and simulations are written in Matlab with the LMI Toolbox to solve the LMI feasability problem (7). Tables 1 to 4 report the area of the beacon intervals found by the algorithm, for the four scenarios respectively. In the following, the simulation results are presented with increasing level of noise. The mapping results are presented in the following order: without noise, with additional white Gaussian noise, with bias, and finally with outliers.

Table 1. Scenario 1 results

Noise	B1 area [m^2]	B2 area [m^2]	B3 area [m^2]	cputime [s]
No noise, 0% outliers	0.171	0.012	0.049	8.4
No noise, 5% outliers	0.171	0.012	0.049	20.5

Table 2. Scenario 2 results

Noise	B1 area [m^2]	B2 area [m^2]	B3 area [m^2]	B4 area [m^2]	cputime [s]
No noise, 0% outliers	0.014	0.014	0.027	0.027	23.8
No noise, 5% outliers	0.014	0.014	0.027	0.027	44.5

Table 3. Scenario 3 results

Noise	B1 area [m^2]	B2 area [m^2]	B3 area [m^2]	B4 area [m^2]	B5 area [m^2]	B6 area [m^2]	cputime [s]
No noise, 0% outliers	0.018	0.018	0.018	0.018	0.037	0.018	816.4
Noise 1, 0% outliers	2.34	0.58	4.69	1.56	2.08	9.37	433.8
Noise 1, 5% outliers	0.29	0.19	0.15	0.29	4.69	0.15	649.9
Noise 1, 10% outliers	0.15	0.29	0.15	0.29	2.34	0.15	743.2
Noise 1, 20% outliers	0.15	0.58	0.15	0.44	0.58	0.15	990.9

Table 4. Scenario 4 results

Noise	B1 area [m^2]	B2 area [m^2]	B3 area [m^2]	B4 area [m^2]	cputime [s]
No noise, 0% outliers	0.015	0.015	00.015	0.046	896.5
Noise 1, 0% outliers	31.64	126.56	126.56	42.19	338.7
Noise 1, 5% outliers	31.64	31.64	63.28	42.19	371.6
Noise 1, 10% outliers	15.82	31.64	63.28	126.56	448.9
Noise 2, 0% outliers	42.19	63.28	253.12	506.25	224.8
Noise 2, 5% outliers	253.12	15.82	42.19	31.64	321.2
Noise 2, 10% outliers	15.82	15.82	10.55	47.46	390.0

4.1 Mapping results without noise

The robot trajectories (black line), true beacon positions (red dots) and corresponding returned boxes (yellow or magenta rectangles) are shown in Fig. 2 and Fig. 3 for Scenarios 1 and 2 respectively. The returned boxes are not visible because they are too small (see the corresponding areas in Table 1) and are hidden by the red dots of the true beacon positions. From Fig. 1 and beacon $B1$, one notices the ring shape of the potential boxes that were explored. This is due to the small size of the trajectory and small number of range measurements. Moreover in the same figure, but for $B2$, one sees that the search was principaly concentrated on two symmetric positions which is commom for range meaurements because of the multi-modality of the constraints. By comparing the two lines of Table 1 and Table 2, we can notice that increasing the percentage of tolerated outliers from 0% to 5% does not reduce the size of the returned boxes. In the next section, we will see that when additional noise corrupts the measurements, a reduction is observed when the percentage is increased.

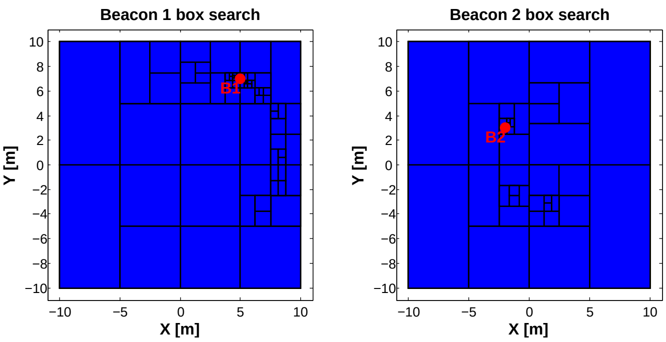

Fig. 1. Scenario 1 (No noise): beacon boxes

4.2 Mapping results with zero-mean white Gaussian noise

Scenarios 3 and 4 were used to test the algorithm against range measurements with zero-mean white Gaussian noise. The results are presented on the rows corresponding to Noise 1 in Tables 3 and 4. By looking at the final box areas (around a few meters square to hundred meters square) for the case of Noise 1 with 0% of outliers tolerated and by considering Fig. 4 and Fig. 5, it seems that the algorithm performs poorly in presence of noise. But in these simulations, no outiers were tolerated. Indeed, by looking at Fig. 6 for Scenario 3 and Fig. 7, Fig. 8 for Scenario 4 where 10% of outliers were tolerated, the returned boxes are smaller, and still contains the true solution. When no outliers are tolerated, the box sizes are larger and even may not contain the true solution (Fig. 4,

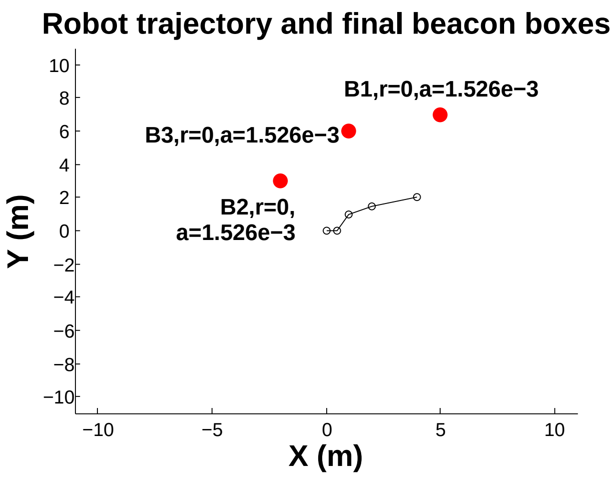

Fig. 2. Scenario 1 (No noise): final (r=% of outliers, a=box area)

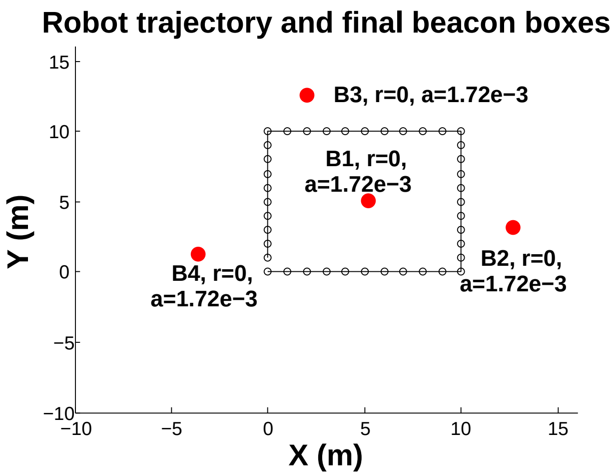

Fig. 3. Scenario 2 (No noise): final (r=% of outliers, a=box area)

*B*4). This result shows that the noise has a significant impact on the box division procedure. Even when the box is still very large, if one of its border is close to the true beacon position, with only one noisy range measurement considered as an outlier, the search can deviated from the true beacon position. This influence is attenuated when some outliers are tolerated.

4.3 Mapping results with bias

It can be seen in the results given in Table 4 and Fig. 9 obtained from Scenario 4 that a bias on the range measurements impacts the size of the returned boxes. As for the case with Gaussian noise, increasing the percentage of tolerated outliers helps to reduce the area of the boxes. However, it can be seen from Fig. 9 that the results of the proposed method are corrupted by a bias, because all the returned intervals are slightly shifted from the true beacon positions.

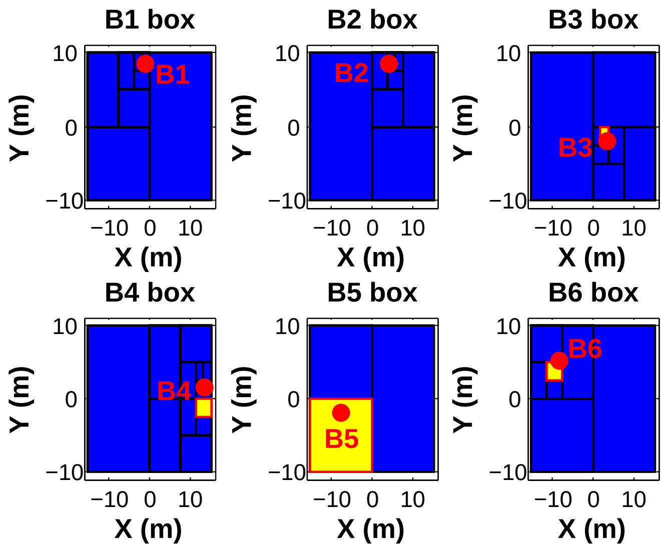

Fig. 4. Scenario 3 (Noise 1, 0% outliers tolerated): beacon boxes (in yellow)

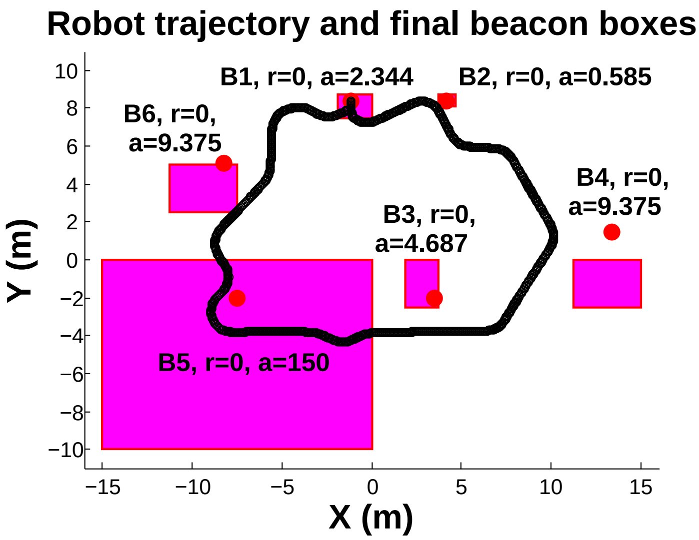

Fig. 5. Scenario 3 (Noise 1, 0% outliers tolerated): final (r=% of outliers, a=box area)

4.4 Mapping results with outliers

In order to evaluate the performance of our method in presence of outliers, we conducted a simulation with Scenario 4 with an increasing percentage of outliers (from 0% to 40%) for each beacon. Moreover, we also compared the results with three standard mapping algorithms. The first one is a trilateration based algorithm (termed Trilat). The second, is a Levenberg-Marquardt nonlinear optimization algorithm (termed LM) which is initialized from the trilateration solution. The third is a version of an occupancy grid algorithm (termed OccGrid). As shown in Fig. 10 for beacon 1, we computed the estimated beacon errors for an increasing percentage of outliers. We can see that both our method (LMI-SoS) and the OccGrid algorithm are

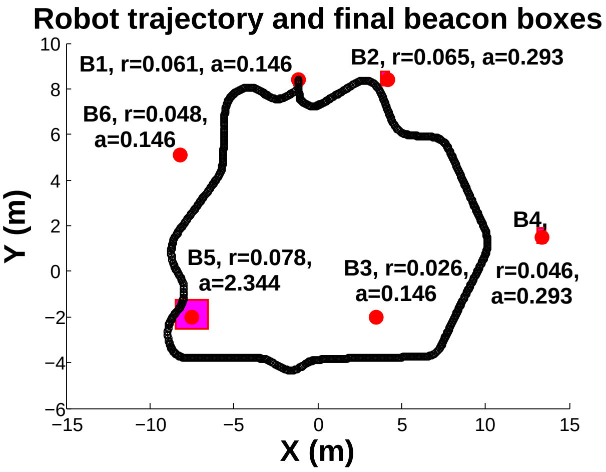

Fig. 6. Scenario 3 (Noise 1, 10% outliers tolerated): final (r=% of outliers, a=box area)

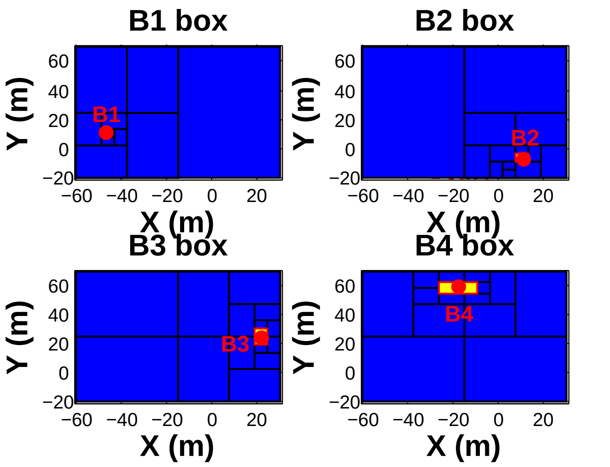

Fig. 7. Scenario 4 (Noise 1, 10% outliers tolerated): beacon boxes (in yellow)

robust to the outliers. Indeed, even if the LM methods gives lower errors with no outliers, the errors quickly grow with the percentage of outliers. The Trilat algorithm shows the same behovior, whereas the OccGrid and the LMI-SoS keep the errors approximately constant. Fig. 11 shows the estimated beacon positions for the four algorithms and for 20% of outliers. The blue rectangles represent the final boxes returned by our LMI-SoS approach. We can see that the final boxes always include the true position, and that the estimated positions provided by OccGrid are also very close. As the Trilat algorithm estimates are far from the true positions, the LM algorithm fails to converge for all beacons. This shows the importance of a good initial estimate for nonlinear optimization. In this case, we could have used the positions estimated by our algorithm to initialize the LM algorithm and obtain a better estimate.

4.5 Discussion

Thereby, with no noise, the boxes have very small areas which in most cases correspond to the threshold area

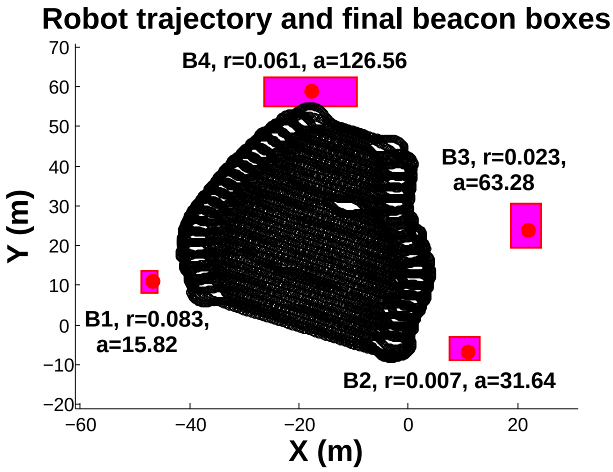

Fig. 8. Scenario 4 (Noise 1, 10% outliers tolerated): final (r=% of outliers, a=box area)

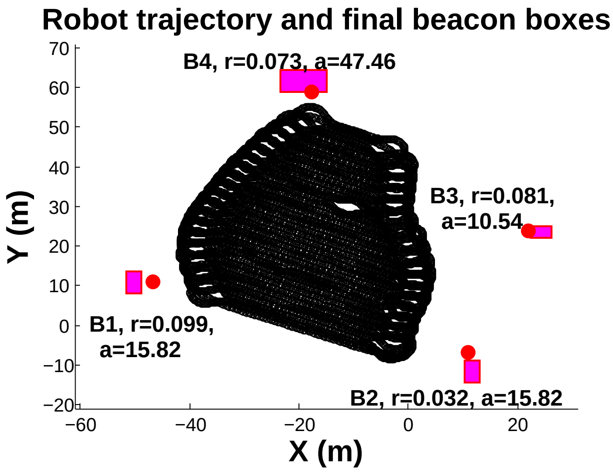

Fig. 9. Scenario 4 (Noise 2, 10% outliers tolerated): final (r=% of outliers, a=box area)

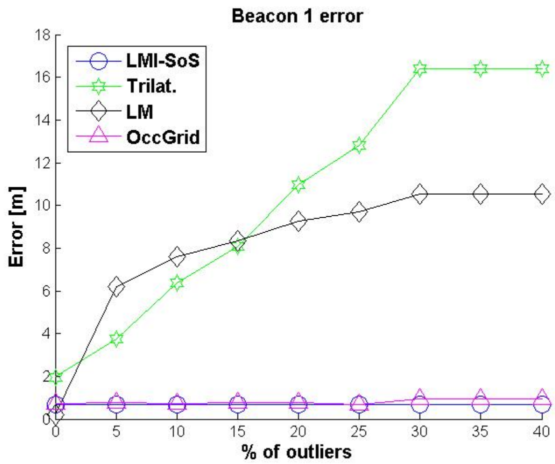

Fig. 10. Position errors of beacon 1 for increasing percentage of outliers

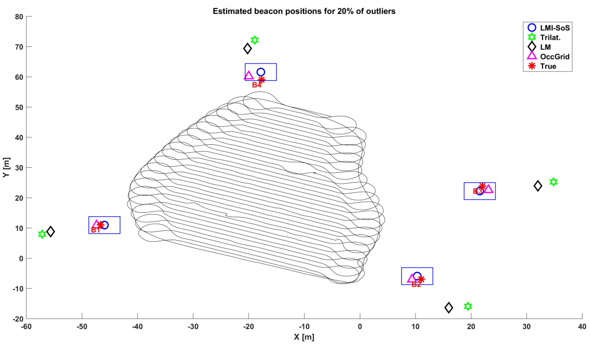

Fig. 11. Estimated beacon positions with 20% of outliers

δ_{area} where the algorithm is stopped. However, in presence of noise, the returned boxes are much larger. To handle correctly the noise, it is mandatory to tolerate a certain amount of outliers, otherwise the returned solutions can be totally incorrect. Because no noise model is used, the method does not seem suitable in the case of biases on the measurements. The main advantage of the proposed method, its robustness, was demonstrated with an increasing percentage of outliers. Compared to other standard methods, it was not affected by outliers and always returned a box enclosing the true beacon position. Finally by looking at the processing times in Tables 1 to 4, it appears that the method is relatively slow and only suited for offline applications. Indeed for each box and each range measurement, a LMI problem has to be solved. Indeed, the BFS algorithm used to find the smallest boxes is very costly and we plan to find an alternative in future works.

5. CONCLUSION

This work presented a new approach for the mapping of beacons given known robot positions, and the range measurements as the only source of information. A LMI feasability test was derived to check if a range measurement is consistent with a box enclosing the true beacon. Using all the range measurements coming from a beacon, the proposed consistency test allows to discard regions of the search space incompatibles with one or a percentage of the measurements. Starting from an initially large box, and employing a breadth-first search algorithm, it is possible to return the smallest box consistent with all the constraints. The proposed algorithm was validated on different simulations with increasing levels of noises and showed promising performances. Indeed, the algorithm demonstrates its ability to handle noisy measurements and outliers by increasing a threshold on the percentage of tolerated outliers. By doing so, the size of the returned boxes where reduced and the estimation of the beacon positions was thus improved. One of the major asset of the proposed algorithm is its robustness to outliers. In future work, we plan to adapt the algorithm for the localization and SLAM problems, or to integrate it as an outlier removal module.

REFERENCES

Blanco, J.L., Fernandez-Madrigal, J.A., and Gonzalez, J. (2008). Efficient probabilistic Range-Only SLAM. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

Boots, B. and Gordon, G.J. (2012). A spectral learning approach to range-only slam. *arXiv preprint arXiv:1207.2491*.

Caballero, F., Merino, L., and Ollero, A. (2010). A general gaussian-mixture approach for range-only mapping using multiple hypotheses. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*.

Di Marco, M., Garulli, A., Giannitrapani, A., and Vicino, A. (2004). A set theoretic approach to dynamic robot localization and mapping. *Autonomous Robots*, 16(1), 23–47.

Djugash, J., Hamner, B., and Roth, S. (2009). Navigating with ranging radios: Five data sets with ground truth. *Journal of Field Robotics*, 26(9), 689–695.

Djugash, J. and Singh, S. (2009). A robust method of localization and mapping using only range. In *Experimental Robotics*, 341–351. Springer.

Fabresse, F., Caballero, F., Maza, I., and Ollero, A. (2013). Undelayed 3d ro-slam based on gaussian-mixture and reduced spherical parametrization. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

Gahinet, P., Nemirovski, A., Laub, A.J., and Chilali, M. (1995). *LMI Control Toolbox*. The MathWorks Inc.

Jaulin, L. (2009). A nonlinear set membership approach for the localization and map building of underwater robots. *IEEE Trans. on Robotics*, 25(1), 88–98.

Jaulin, L. (2011). Range-only slam with occupancy maps: A set-membership approach. *IEEE Trans. on Robotics*, 27(5), 1004–1010.

Lasserre, J.B. (2007). A sum of squares approximation of nonnegative polynomials. *SIAM Review*, 49(4), 651–669.

Lee, C. (1961). An algorithm for path connections and its applications. *IRE Trans. on Electronic Computers*, EC-10(3), 346–365.

M.D. Choi, T.Y. Lam, B.R. (1995). Sums of squares of real polynomials. 58, 103–126.

Moravec, H. and Elfes, A. (1985). High resolution maps from wide angle sonar. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*.

Nesterov, Y. and Nemirovski, A. (1994). *Interior-point Polynomial Methods in Convex Programming*. SIAM.

Olson, E., Leonard, J.J., and Teller, S. (2006). Robust range-only beacon localization. *IEEE Journal of Oceanic Engineering*, 31(4), 949–958.

Pani Paudel, D., Habed, A., Demonceaux, C., and Vasseur, P. (2015). Robust and optimal sum-of-squares-based point-to-plane registration of image sets and structured scenes. In *IEEE Int. Conf. on Computer Vision (ICCV)*.

Powers, V. and Wormann, T. (1998). An algorithm for sums of squares of real polynomials. *Journal of Pure and Applied Algebra*, 127(1), 99–104.

Zhou, Y. (2009). An efficient least-squares trilateration algorithm for mobile robot localization. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.