

Limits of Trilateration-Based Sensor Placement Algorithms

Lionel G  n  v  
ICube Laboratory
University of Strasbourg-CNRS
Strasbourg, France
Email: l.geneve@unistra.fr

Olivier Kermorgant
LS2N
  cole Centrale de Nantes
Nantes, France
Email: olivier.kermorgant@ec-nantes.fr

  douard Laroche
ICube Laboratory
University of Strasbourg-CNRS
Strasbourg, France
Email: laroche@unistra.fr

Abstract—To localize a mobile vehicle in a predefined area, it is possible to resort to a positioning system using beacons. The questions arising then are: what is the right number of beacons to deploy, and where should they be positioned to accurately localize the target. With a sensor placement algorithm, it is possible to generate a configuration of the beacons that optimize some quality criteria. However, designing the objectives to optimize is not simple, and the solution of the sensor placement algorithm may not always ensures a good localization in practice. This work evaluates the impact of the objectives of a sensor placement algorithm on different localization techniques. We show that criteria based on trilateration are not sufficient anymore when the localization is done with other techniques than trilateration. Indeed, data fusion-based localization algorithms use additional information such as odometry, and are less sensitive to poor coverage or poor beacon configurations than trilateration. Thus, designing new criteria taking into account the dynamics of the vehicle would probably improve further the placements and use less beacons for the same performance.

I. INTRODUCTION

In mobile robotics, it is very common to localize a robot using a set of static sensors in the environment. These sensors or beacons generally provide two kinds of information: bearing and range. In this work, we consider the case of range-only beacons. These past few years, there has been some major improvements in ranging devices by trying to find an alternative to the Global Positioning System (GPS) which is limited for open-sky outdoor situations. A promising and emerging technology is the use of Ultra Wide Band radio-frequencies [1], that allows accurate time-of-flight range measurements. This type of ranging sensor gathers some interesting characteristics for the localization of mobile devices: a precision around ten centimeters, a maximum range of several tens of meters, and the ability of uniquely identify each beacon, thus avoiding the data association problem. In this context, we consider the localization of a mobile vehicle with a set of static sensors providing range measurements. This could be for instance the case of a mobile robot deployed in a warehouse for surveillance or to perform some automatic tasks. One aim is to use the least possible number of sensors for cost and setup reasons. With a sensor placement algorithm, it is possible to obtain the number of sensors necessary to cover the area of interest, along with their positions. The algorithm is composed of two main components: the optimization framework and

the objectives or criteria to optimize. In this work, a genetic algorithm (GA) will be used to solve the optimization problem. As it is commonly done, the fitness function, which is composed from a single or multiple criteria, will be designed using a trilateration-based localization method. However, we will show that trilateration-based criteria are not well adapted when the mobile vehicle is localized by an extended Kalman filter (EKF), which is a standard localization technique. Indeed, generally the mobile vehicle is equipped with different sensors and their information are fused to obtain an estimate of the robot state. The main advantage over trilateration is that the robot can rely on proprioceptive sensors and perform dead-reckoning for a short period when there is no or few external measurements. A downside of the trilateration is indeed that it can only be performed when at least three range measurements are available.

The rest of this paper is organized as follows. In the next section, previous works on sensor placement algorithms which mostly rely on a trilateration-based criterion are presented. In section III, our GA-based optimization framework is presented with the fitness functions. Section IV presents the performance of the sensor placement algorithm on different localization algorithms. Finally, section V concludes and proposes an outline of how the dynamics of the vehicle being localized can be integrated in the fitness function.

II. RELATED WORKS

In order to solve the sensor placement problem, it is necessary to define what are the objectives to optimize. Laguna et al. identified three characteristics [2]:

- Minimize the number of beacons,
- Maximize the coverage of the robot's workspace,
- Maximize the percentage of the area that is covered with admissible dilution of precision (DOP) values.

The last point can be reformulated more generally as: maximize the localization accuracy on the entire workspace. Most of the time, the last two objectives are based on the static trilateration method. Allen et al. [3] proposed a localization utility function, based on the number of range measurements available at a given position. The more range measurements, the better the localization is, and ideally we seek to have at least three range measurements at each positions to be able

to perform trilateration. In Burke et al. [4], the range measurement uncertainties are propagated through the trilateration equations to get the estimated position uncertainty. The cost function is defined as the mean over the environment of an error metric, which is defined as the square root of the sum of the eigenvalues of the position uncertainty. A Nelder-Mead downhill simplex method is used to solve the non-convex optimization problem. Other localization accuracy metrics have been used as objectives in the fitness function. Among them, the geometric dilution of precision (GDOP) [5], and the Fisher Information Matrix (FIM) [6]. These metrics provide an information on the effects of the geometric configuration of the beacons on the location estimate. Kirchhof [7] used the GDOP with binary and mixed integer programming, and nonlinear programming to solve the optimization. Roa et al. [8] combined the DOP with a non-availability term to create a weighted fitness function of two terms. The first one is the mean of the DOP values over the workspace, and the second one is the percentage of area where no range measurement is available. The problem is solved using a meta-heuristic optimization strategy called diversified local search. The FIM has also been extensively used with different formulations. Jourdan et al. [9] used a cost function based on the square root of the trace of the FIM inverse and a coordinate-descent algorithm. Most of the other works uses the FIM determinant as a basis for the cost function [6], [10], [11], [12]. The FIM determinant expression was derived in 2D and 3D and for different range measurement modelizations.

Most of the cited works are based on a fitness function with objectives related to the trilateration algorithm. Indeed, the computation of the FIM determinant requires at least three range measurements as for trilateration, which makes it impossible to compute when this condition is not met. Thus, when there are only one or two range measurements, this may results in a loss of information during the optimization procedure. Although it is commonly used to localize mobile devices with only range measurements, the trilateration algorithm comes with several drawbacks. The first one, is that at least three measurements are necessary to perform the computation and thus imposes a high density of beacons in the workspace to avoid non-covered areas. Moreover, it only relies on one source of information. In practice, a mobile robot is often equipped with multiple sensors providing different informations (wheel encoders, IMU, laser, camera, etc). For this reason, the mobile robot is generally localized with an algorithm able to fuse all these information such as the EKF [13] or the particle filter [14]. In this work, it will be shown that indeed the benefit of the beacon placement is not the same in practice when localizing the mobile device by a trilateration or a filtering technique. This raises the question of the design of new objectives to minimize that take into account the dynamics of the mobile robot and the additionnal sensor information. The next section presents the algorithm developed to compute the best positions for the beacons.

III. SENSOR PLACEMENT ALGORITHM

In this section, the framework used to find the optimal sensor placement is described. First, some notation and implementation details are given. Then, the genetic algorithm employed is briefly described, and finally the fitness functions being optimized are presented.

A. Problem formulation

We consider the problem of positioning a set of N static beacons in order to localize a mobile robot operating in an outdoor and closed environment. The problem is expressed in 2D by considering that the beacons and the robot transmitters are at the same height, or that the heights are known. We assume that the beacons can only be positioned at the border of the workspace, or on the borders of any forbidden region lying inside the workspace (for example a building in a parking lot). A beacon position is given by $\mathbf{p}_i = (x_i, y_i)^T$, $i = 1, \dots, N$, but in the optimization it is parameterized by a curvilinear abscissa $s_i \in [0, 1[$ along the boundaries. This reduces the search space from two to one, but introduces spatial discontinuities when passing from the workspace boundaries to the first obstacle and from one obstacle to another. To evaluate the fitness functions, the workspace is discretized into a squared 2D grid of K cells. The size of the cells depends on the range measurement accuracy. A position in the workspace is given by $\mathbf{p}_k = (x_k, y_k)^T$, $k = 1, \dots, K$. The beacons provide range measurements, modeled by:

$$r_{i,k} = d_i + \epsilon_{i,k}$$

where $d_i = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$ is the true distance, and $\epsilon_{i,k} \sim \mathcal{N}(0, \sigma_i^2)$ is a white Gaussian noise, with σ_i^2 the distance-dependent variance. Using the formulation given in [11] the expression of σ_i^2 is given by: $\sigma_i^2 = (1 + \eta)^2 \sigma_{i,0}^2$, with η the distance-dependent parameter. As only one type of technology is used for the beacons, we assume that $\sigma_{i,0}^2 = \sigma_r^2$ is the same for all beacons. The beacons also have a maximal operational range r_{max} . We made the hypothesis that only clear line-of-sight (LOS) range measurements can be used to compute a location estimate. In practice, non-LOS measurements are possible with RF beacons, but the effects of the obstacle on the measurement are not simple to model and we choose to simply discard them. Moreover, when the localization is done with an EKF, outliers can be automatically discarded using a Mahalanobis distance test.

B. Genetic algorithm

The optimization is solved using a genetic algorithm [15]. The number of individuals in the population is fixed to 100. The first population is initialized with random placements except for one that is uniform. The algorithm is decomposed in three steps: the selection, the crossing and the mutation. The selection implementation follows an elitism and tournament principle. The crossing operation randomly selects two parents to create a new child. Finally, mutations are randomly affecting some individuals slightly modifying the curvilinear abscissa of one of the beacons. The algorithm is stopped if one of the

two conditions is met: the number of iteration reaches 200, or the best cost has not changed during 10 consecutive iterations. Before the selection step, the individuals are evaluated through a fitness function, indicating how well adapted they are with respect to the objectives. The next section presents the fitness functions used in this work in more details.

C. Fitness functions

Once the optimization framework has been defined, the next step is to find a suitable fitness function to score and rank the individuals. Three fitness functions are considered, all based on existing works but with a slightly different formulation. The first one (termed nbRanges) is related to the coverage [3], and computes the mean number of range measurements available at each position of the workspace. The second (termed detFimThresh), is related to the localization accuracy, and uses the FIM determinant ($|FIM|$) as in [11]. Precisely, it computes the percentage of the workspace where: $|FIM| \geq t$, with t a threshold controlling the desired localization accuracy. Indeed the higher is $|FIM|$, the better the localization accuracy. The third fitness function (termed mixedCosts) follows the idea of [8] and combines the coverage term and the localization accuracy term that have been introduced previously. The first term is weighted by 0.7, and the second by 0.3. The coverage term has a higher weight because it is prerequisite for the computation of the localization accuracy. Now that the optimization framework and the fitness functions have been defined, we present the simulation results that compare the localization performances depending on the placement of the beacons and on three localization methods.

IV. SIMULATION RESULTS

As the purpose of the beacons is to localize a mobile robot, the performances of the configurations given by the GA are tested on the localization accuracy of a differential-drive mobile robot following a reference trajectory. Three algorithms are considered to localize the mobile robot: trilateration [4] (termed Trilat.) and two versions of an EKF. The first EKF version (termed T-EKF) uses the trilateration solution for the correction step, whereas the second (termed R-EKF) directly fuses the range measurements. Both EKF use the odometry given by the robot wheel encoders for the prediction step. We also compare the placements and localization results given by the GA with a uniform placement (UNIF) of the beacons to validate the interest of the optimization. The metric used to evaluate the accuracy of a localization algorithm and its corresponding beacon configuration is the root mean square error between the reference and estimated trajectories. For each beacon placement, 200 runs with different Gaussian noises on the odometry and range measurements are generated and the estimated trajectory computed by the localization algorithms. The average trajectory errors, standard deviations, minimum and maximum errors are then computed. The different parameters used for the simulations are $\sigma_r^2 = 1.1e^{-3} \text{ m}^2$ the initial variance of the range measurements, $\eta = 0.05$ the distance-dependent parameter, $r_{max} = 10 \text{ m}$ the maximal range, and

$t = 1296$ the threshold on the $|FIM|$ corresponding to a desired localization accuracy of $\pm 0.5 \text{ m}$. For the odometry measurements, both wheel velocities are affected by a white Gaussian noise with a variance of $1.0e^{-6} \text{ m}^2\text{s}^{-2}$ and we also introduce a small bias by setting one wheel bigger than the other by 0.5 mm . In the rest of this section, we first compare the performances of the sensor placement algorithm to the uniform placement. Then, using the beacon configurations given by the GA with the different fitness functions, we compare the performances of the three localization algorithms.

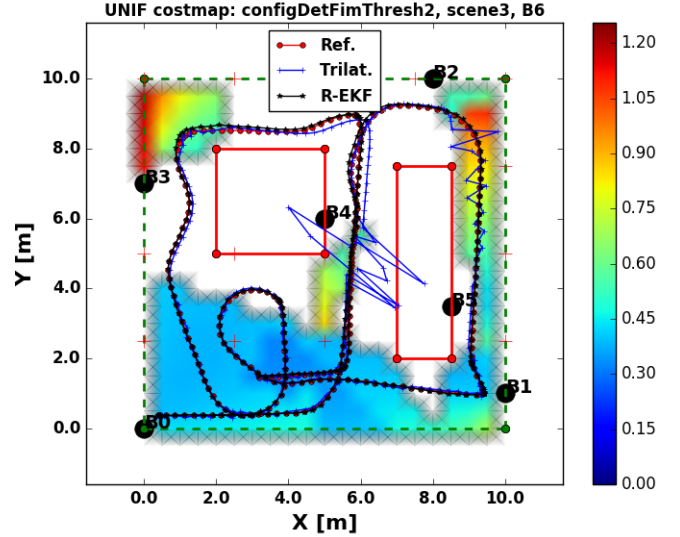


Fig. 1. Cost map for the uniform placement for 6 beacons, evaluated with the DetFimThresh fitness function (the colors correspond to the localization accuracy: a value d from the scale corresponds to a localization accuracy within $\pm d$, thus areas with good localization are in blue). With the uniform placement, most of the workspace is filled in white which corresponds to positions where no position information is available (there is less than three range measurements). The blue line corresponds to the robot's trajectory estimated by the trilateration algorithm, whereas the black one corresponds to the robot's trajectory estimated by the R-EKF algorithm, the red line is the ground truth trajectory.

A. Sensor placement evaluation

The workspace used for the simulations is shown in Fig. 1 and Fig. 2. The figures display the estimated localization accuracy obtained after a uniform placement (Fig. 1) and a GA placement with the DetFimThresh fitness function (Fig. 2). The color filling the workspace corresponds to the localization accuracy ($\pm d$). As the objective is to cover the entire workspace with a given localization accuracy (here $d = 0.5 \text{ m}$), we want to fill as much as possible the workspace in blue or cyan (which correspond to a localization better than $\pm d = 0.5 \text{ m}$). The white areas correspond to positions where there are less than three range measurements available (and thus no position estimate can be computed from the trilateration algorithm). By comparing both figures, we see that the sensor placement is actually increasing the coverage of the area and that with the GA-DetFimThresh placement, most of

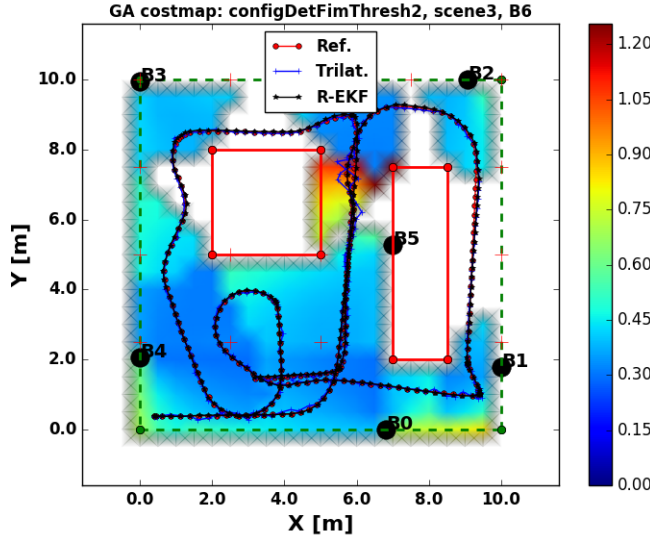


Fig. 2. Cost map for the GA placement for 6 beacons, evaluated with the DetFimThresh fitness function (the colors correspond to the localization accuracy: a value d from the scale corresponds to a localization accuracy within $\pm d$, thus areas with good localization are in blue). With the GA-DetFimThresh placement, most of the workspace is filled in cyan which corresponds to a localization precision below the desired threshold of ± 0.5 m. Notice also that the white areas (with no positions information) are smaller. The blue line corresponds to the robot's trajectory estimated by the trilateration algorithm, whereas the black one corresponds to the robot's trajectory estimated by the R-EKF algorithm, the red line is the ground truth trajectory.

the workspace is filled with blue or cyan. On the contrary, for the uniform placement, most of the top left side is not covered by at least three range measurements (white regions) and thus no position information will be available from the trilateration localization algorithm. The same statement can be done for the NbRanges fitness function by comparing Fig. 3 and Fig. 4. In these two figures, the colors correspond to the number of range measurements available. We can see that after the GA optimization most of the workspace is covered by at least three range measurements (red) except for two small regions (yellow and cyan). Whereas for the uniform placement, almost half of the workspace is still covered by only one (cyan) or two (yellow) range measurements.

Fig. 1 shows that in the areas with poor localization accuracy (yellow and red areas in the middle and top right corner) the trajectory estimated by trilateration (blue line) is oscillating. The reason is that in these areas the trilateration is done using beacon configurations where the beacons are nearly aligned, which is far from perfect and known to perform poorly. The same effect can be observed in Fig. 4.

In Fig. 5 we see that the uniform placement, whatever the localization method, gives most of the time the largest mean trajectory errors compared to the optimal placements obtained with the different fitness functions. This is especially true when the number of beacons is small, as it can be seen from Fig. 6 where the performances of the uniform and GA-

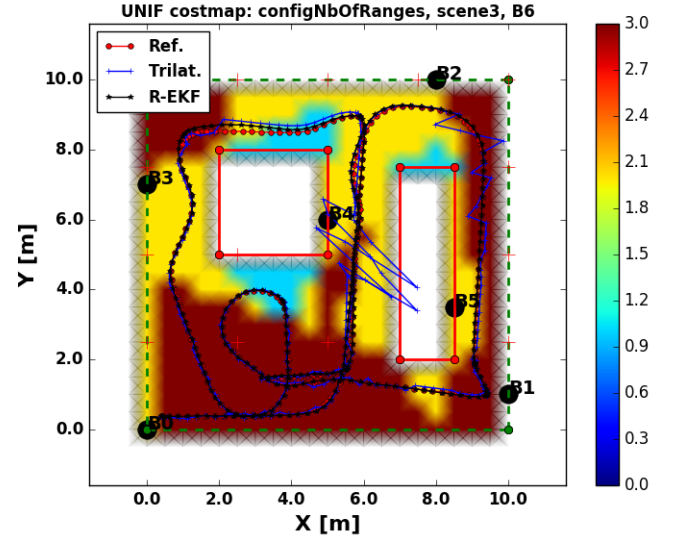


Fig. 3. Cost map for the uniform placement for 6 beacons, evaluated with the NbRanges fitness function (the colors correspond to the number of range measurements available: cyan=1, yellow=2, red=3 or more). The blue line corresponds to the robot's trajectory estimated by the trilateration algorithm, whereas the black one corresponds to the robot's trajectory estimated by the R-EKF algorithm, the red line is the ground truth trajectory.

DetFimThresh placements are compared. When the number of beacons increases, the gap between the uniform and the GA placements decreases, thus limiting the interest of the beacon placement algorithm. Finally, among the fitness functions used for the GA placements, DetFimThresh seems to give better results than MixedCosts and NbRanges, respectively. This tendency, can be observed in Fig. 5, where the errors are decreasing from left to right when comparing the four placements.

B. Localization performances

TABLE I
LOCALIZATION ERRORS FOR THE 3 LOCALIZATION ALGORITHMS WITH A UNIFORM PLACEMENT AND 6 BEACONS

| Localization method | Mean error [m] | Std error [m] | Max error [m] | Min error [m] |
|---------------------|----------------|---------------|---------------|---------------|
| Trilat. | 0.549 | 0.210 | 1.603 | 0.271 |
| T-EKF | 0.194 | 0.056 | 0.465 | 0.118 |
| R-EKF | 0.081 | 0.014 | 0.112 | 0.040 |

TABLE II
LOCALIZATION ERRORS FOR THE 3 LOCALIZATION ALGORITHMS WITH THE FITNESS FUNCTION NbRanges AND 6 BEACONS

| Localization method | Mean error [m] | Std error [m] | Max error [m] | Min error [m] |
|---------------------|----------------|---------------|---------------|---------------|
| Trilat. | 0.766 | 0.028 | 0.858 | 0.704 |
| T-EKF | 0.219 | 0.050 | 0.391 | 0.147 |
| R-EKF | 0.046 | 0.007 | 0.064 | 0.029 |

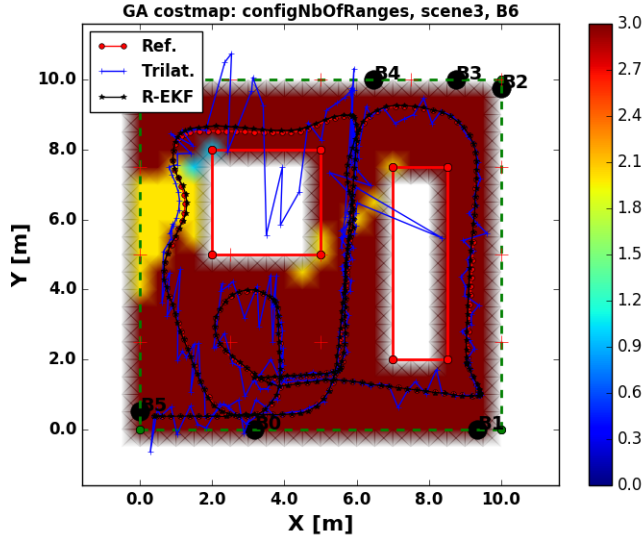


Fig. 4. Cost map for the GA placement for 6 beacons, evaluated with the nbRanges fitness function (the colors correspond to the number of range measurements available: cyan=1, yellow=2, red=3 or more). The blue line corresponds to the robot's trajectory estimated by the trilateration algorithm, whereas the black one corresponds to the robot's trajectory estimated by the R-EKF algorithm, the red line is the ground truth trajectory.

TABLE III
LOCALIZATION ERRORS FOR THE 3 LOCALIZATION ALGORITHMS WITH THE FITNESS FUNCTION MIXEDCOSTS AND 6 BEACONS

| Localization method | Mean error [m] | Std error [m] | Max error [m] | Min error [m] |
|---------------------|----------------|---------------|---------------|---------------|
| Trilat. | 0.47 | 0.017 | 0.515 | 0.437 |
| T-EKF | 0.154 | 0.050 | 0.410 | 0.091 |
| R-EKF | 0.041 | 0.006 | 0.061 | 0.028 |

Tables I to IV summarize the trajectory errors obtained from the three localization algorithms, for six beacons and for the uniform, GA-NbRanges, GA-MixedCosts, and GA-DetFimThresh placements respectively. By comparing the tables, we can notice that the mean trajectory errors are always lower for the R-EKF localization. Also, most of the time, the T-EKF algorithm gives better results than the pure trilateration method. As this is quite obvious, knowing that the EKF's use additional information from the odometer sensors, it also reveals that the trilateration-based objectives used in the fitness functions are not optimal for other localization algorithms. Using a fitness function based on trilateration for the sensor placement optimization improves mostly the localization by trilateration, but has little or less impact on EKF based localization algorithms. This can also be seen in Fig. 5, where the localization errors obtained with the trilateration algorithm (red bars) are reduced a lot compared to the uniform placement and the GA-MixedCosts and GA-DetFimThresh placements. However, for the EKF algorithms (green and blue bars), the errors are still lower with the optimized placement compared to the uniform one, but the gap is smaller.

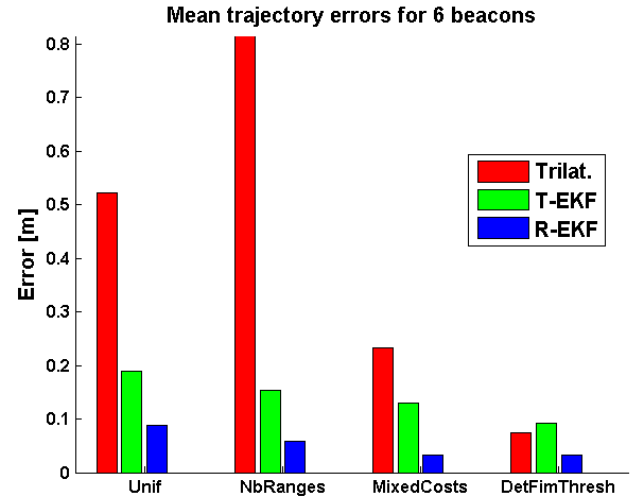


Fig. 5. Comparison of the mean trajectory errors for the 4 placements and 3 localization algorithms for 6 beacons. The Trilat. localization errors are greatly reduced between the uniform placement and the GA-MixedCosts and GA-DetFimThresh (the height of the red bars is divided by two and five). However for the EKF's (green and blue bars), the reduction of the errors between the uniform and optimized placement is not as important. Thus, the Trilat. localization benefits from the sensor placement algorithm, but the EKF's less.

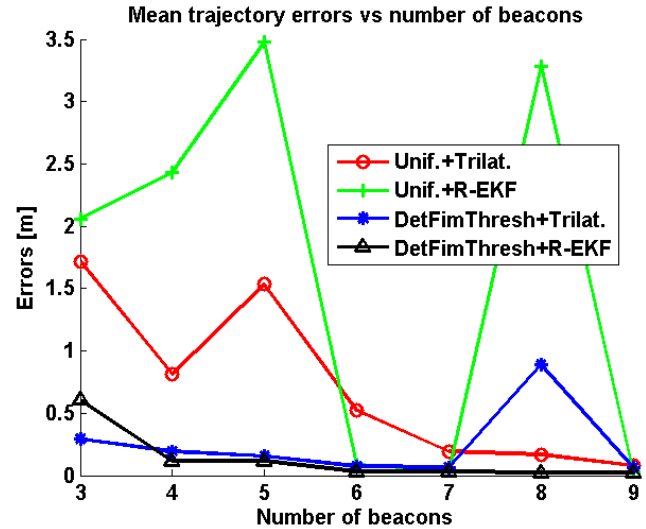


Fig. 6. Evolution of the mean trajectory errors for the uniform and DetFimThresh placements and for the Trilat. and R-EKF localization algorithms

From Fig. 6, which shows the evolution of the mean trajectory errors when the number of beacons increases, we can see that with the GA-DetFimThresh placement, the trilateration errors (blue line) are higher than the R-EKF errors (black line), except with three beacons. Thus, for a given mean trajectory error, it is often necessary to use one or two additional beacons with the trilateration algorithm to obtain the same results as with the R-EKF algorithm. This suggests, that when employing more appropriate fitness functions during

TABLE IV
LOCALIZATION ERRORS FOR THE 3 LOCALIZATION ALGORITHMS WITH
THE FITNESS FUNCTION DETFIMTHRESH AND 6 BEACONS

| Localization method | Mean error [m] | Std error [m] | Max error [m] | Min error [m] |
|---------------------|----------------|---------------|---------------|---------------|
| Trilat. | 0.071 | 0.004 | 0.087 | 0.063 |
| T-EKF | 0.094 | 0.009 | 0.117 | 0.074 |
| R-EKF | 0.030 | 0.004 | 0.042 | 0.022 |

the beacon placement optimization, we could use even less beacons for the localization.

The trilateration algorithm has two main drawbacks compared to the EKF. The first one is that when there are less than three range measurement available, no estimated positions can be computed, which leaves the robot completely blind during some period. The second is that the algorithm is sensitive to areas of poor localization accuracy due to degenerate beacon configurations. The positions estimated by trilateration can then present large errors. Regarding the EKF, these two situations have less impact. First, the EKF algorithm can rely for short periods on the odometry to localize the robot when there is no range measurement available. Second, the R-EKF does not require three measurements to make a correction, this step can be performed with one or more measurements. These situations are illustrated in Fig. 1 and Fig. 2, where the trajectories estimated by trilateration are represented with a blue line, whereas the R-EKF estimated trajectories are in black (the ground truth trajectory is in red). As can be seen, the trajectory estimated by the R-EKF is always very close to the ground truth trajectory, whereas the trajectory estimated by trilateration is sometimes oscillating and drifting. This clearly shows the limitations of the current fitness functions which are designed based on a static trilateration algorithm. Indeed, in these formulations, the dynamic (odometry) of the robot, and the ability to localize and fuse the range measurements even with less than three at each timestep are not taken into account.

V. CONCLUSION

We presented a sensor placement algorithm using different trilateration-based fitness functions and optimized with a genetic algorithm. We highlighted the fact that these fitness functions were not well adapted when the mobile vehicle localizes itself with an algorithm using additional information such as odometers. The trilateration and trilateration-based extended Kalman filter (EKF) algorithms are more affected by non-covered areas and poor sensor configurations than a standard EKF fusing the range measurements directly. This motivates the fact of designing a novel fitness function taking into account the benefits of more sophisticated localization methods to optimize further the placement of the sensors.

Especially, the number of sensors used for localization could be reduced. Knowing that a Kalman filter can rely for short time periods on the odometry without any external range measurements, it would be possible to relax the full workspace coverage condition and obtain a better distribution of the beacons. One lead for the design of the novel fitness function would be to, similarly on what is done with the Fisher information matrix, derive a criterion on the innovation or covariance matrix of the EKF. Future works will be carried on the design of a fitness function suited for the EKF localization.

REFERENCES

- [1] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, July 2005.
- [2] M. Laguna, J. O. Roa, A. R. Jimenez, and F. Seco, "Diversified local search for the optimal layout of beacons in an indoor positioning system," *IEEE Transactions*, vol. 41, no. 3, pp. 247–259, 2009.
- [3] R. Allen, N. MacMillan, D. Marinakis, R. Nishat, R. Rahman, and S. Whitesides, "The range beacon placement problem for robot navigation," in *Canadian Conference on Computer and Robot Vision (CRV)*, 2014.
- [4] M. Burke and N. Bos, "Optimal placement of range-only beacons for mobile robot localisation," in *4th Robotics and Mechatronics Conference of South Africa (RobMech)*, 2011.
- [5] R. Yarlagadda, I. Ali, N. Al-Dhahir, and J. Hershey, "Gps gdop metric," *IEEE Proceedings Radar, Sonar and Navigation*, vol. 147, no. 5, pp. 259–264, Oct 2000.
- [6] S. Martinez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.
- [7] N. Kirchhof, "Optimal placement of multiple sensors for localization applications," in *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, 2013.
- [8] J. Roa, A. Jimenez, F. Seco, J. Prieto, and J. Ealo, "Optimal placement of sensors for trilateration: Regular lattices vs meta-heuristic solutions," in *Computer Aided Systems Theory EUROCAST*, ser. Lecture Notes in Computer Science, R. Moreno Diaz, F. Pichler, and A. Quesada Arens, Eds. Springer Berlin Heidelberg, 2007, vol. 4739, pp. 780–787.
- [9] D. Jourdan and N. Roy, "Optimal sensor placement for agent localization," in *IEEE/ION Position, Location, And Navigation Symposium*, 2006.
- [10] A. N. Bishop, B. Fidan, B. D. Anderson, K. Dogancay, and P. N. Pathirana, "Optimality analysis of sensor-target localization geometries," *Automatica*, vol. 46, no. 3, pp. 479–492, 2010.
- [11] D. Moreno-Salinas, A. Pascoal, and J. Aranda, "Optimal sensor placement for underwater positioning with uncertainty in the target location," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.
- [12] J. Perez-Ramirez, D. Borah, and D. Voelz, "Optimal 3-d landmark placement for vehicle localization using heterogeneous sensors," *IEEE Trans. on Vehicular Technology*, vol. 62, no. 7, pp. 2987–2999, Sept 2013.
- [13] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadcopter state estimation," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.
- [14] J. Gonzalez, J. Blanco, C. Galindo, A. O. de Galisteo, J. Fernandez-Madrigal, F. Moreno, and J. Martinez, "Mobile robot localization based on ultra-wide-band ranging: A particle filter approach," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 496–507, 2009.
- [15] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic programming: an introduction*.