# A Hybrid Filter-based and Graph-based approach to SLAM

Francesco Conte, Agostino Martinelli

# A Hybrid Filter-based and Graph-based approach to SLAM

Francesco Conte and Agostino Martinelli

*Abstract*— **This paper introduces a new approach to SLAM which combines an Information Filter and a non linear optimizer. The basic idea of the suggested technique is to use the Information Filter when the system non linearities are negligible, and to switch to the use of the non linear optimizer when the non linearities are not negligible. Extensive simulations are provided in order to evaluate the performance of the proposed approach. In particular, a comparison with the Exactly Sparse Delayed-state Filers (ESDF) technique is carried out.**

## I. INTRODUCTION

In the Simultaneous Localization and Mapping (SLAM) problem, a mobile robot has to be able to autonomously explore an unknown environment with its on-board sensors, gain knowledge about it, interpret the scene, build an appropriate map and localize itself relative to this map. One of the most popular approaches to SLAM is the Filter-based approach, in which the robot kinematics and observations are modeled by a non linear system and tracked via an Extended Kalman Filter (from now on EKF).

In 2006 Eustice *et al,* ([3]) introduced the innovative technique *Exactly Sparse Delayed-State Filters* (from now on ESDF). Basing on the Information Filter (from now on IF or EIF with linearization) and only introducing negligible approximations on the the state recovery, this technique solves the SLAM problem through a constant-time filtering algorithm. This means that the ESDF computational cost does not grow up with the environment size. Therefore, ESDF is considered as the solution to the scalability problem for arbitrately large environments.

Because of this optimal computational behaviour, the ESDF cannot be improved in terms of computational cost through a Filter-based solution to SLAM. On the other hand, the EIF used by the ESDF method suffers from the same limitation of the EKF-SLAM in terms of map accuracy. In particular, as well as the EKF, the EIF is based on the linear approximation of the analyzed system. This approximation represents the main limitation of every EKF-based (or EIF-based) solutions to SLAM because the robot motion and observations generally have a strong non linear nature. Julier and Uhlmann [5] and Castellanos *et al.* [1] proved that the conventional EKF-SLAM yields an inconsistent map. They pointed out that this inconsistency arises from linearization. Indeed, this approximation only holds if the difference between the estimated state and the ground truth is small. Now, in any map representation, the corresponding robot location will drift if no crucial event like a loop closure occurs. Therefore, when the drift is large enough, the linearization is not a possible approximation. Moreover, such drift will be naturally larger when the size of the environment increases. Hence, the estimation process could become inconsistent if the environment is large enough.

Recently, a new strategy has emerged that offers the possibility to solve the SLAM problem without any linear approximation. This approach is called Graph-based approach. It consists in facing the SLAM as a non linear optimization problem: *find the robot trajectory and the map with the greatest probability, given the sensor measurements*. In the works realized by Olson [8] and Grisetti [4] non linear optimization algorithms are proposed in order to solve the SLAM problem. These suggested algorithms are able to build very accurate maps, with a low computational cost.

In this paper we introduce a new approach to SLAM which combines an EIF and a non linear optimizer. In particular, we suggest a hybrid solution to SLAM which consists in using a suitable modification of the ESDF filtering algorithm when the system non linearities are supposed to be negligible, and switching to a non linear optimizer when the system non linearities are stronger (e.g. loop closure). An analogous strategy was proposed in [6] in the context of the Relative Map Approach to solve SLAM. We point out that the aim of this work is not to use or to improve the optimization algorithms suggested in [8] or in [4], but we only want fuse their basic idea about the SLAM problem with the a more conventional Filter-based solution to SLAM .

This work is organized as it follows. In Section II we introduce the ESDF technique pointing out its advantages and drawbacks. In section III we propose our hybrid solution to SLAM. In section IV extensive simulation results are provided in order to evaluate the performance of the proposed approach. In particular, a comparison with the ESDF is carried out. Conclusions are presented in section V.

## II. ADVANTAGES AND DRAWBACKS IN THE ESDF-APPROACH

In 2006 Eustice *et al.* [3] introduced the innovative technique called *Exactly Sparse Delayed-state Filters*. The ESDF algorithm succeeds in exploiting the benefits of the EIF by maintaining a sparse structure of the information matrix (covariance inverse), without any approximation. This is obtained through a *state-augmentation* technique and yields a constant-time computational cost per iteration. In the

This work was supported by the European Commission FP7-ICT-2007-3.2.2 Cognitive Systems, Interaction, and Robotics under the contract #231855 (sFLY).

F. Conte is with the Dipartimento di Ingegneria Elettrica e dell'Informazione, Facoltà di Ingegneria, Università degli Studi dell'Aquila, Italy e-mail: francesco.conte@univaq.it

A. Martinelli is with INRIA Rhone Alpes, Montbonnot, France e-mail: agostino.martinelli@ieee.org

following we will summarize the ESDF method pointing out some of its key properties.

Let us represent the robot motion and perception by the following equations:

$$x_k = f\left(x_{k-1}, u_k + w_k\right) \tag{1}$$

$$z_k = h(x_k, m) + v_k \tag{2}$$

where $x_k$ is the robot pose at the time step $k$, $u_k$ is the control input (proprioceptive measurement), $z_k$ is the exteroceptive measurement available at the time step $k$, $m$ is the environment map, $f$ is the robot motion function, $h$ is the observation function, and $w_k$ and $v_k$ are the proprioceptive and exteroceptive measurements errors, respectively.

The ESDF key idea is to extend the estimated state vector each time an observation occurs. Specifically, at the time step $k$ the current estimated vector is:

$$X_k^T = \begin{pmatrix} x_k^T & M^T \end{pmatrix} \tag{3}$$

where $M$ is a vector carrying all the maintained old poses and the map $m$. In the following we will often talk about the size of $X_k$ as the environment size.

The ESDF method uses the *information form* in order to represent the Gaussian distribution. In this representation the conventional pair $(\mu_k, \Sigma_k)$, mean and covariance value for the state $X_k$, is replaced with the pair $(\eta_k = \Sigma_k^{-1}\mu_k, \Lambda_k = \Sigma_k^{-1})$, corresponding information vector and matrix.

Basing on the information form and the state augmentation, the ESDF technique solves the SLAM problem by performing the following tasks: *motion update*, *state augmentation* and *observation update*. If we suppose that the current state mean $\mu_k$ is available at each iteration (i.e. the state recovery problem is supposed to be solved), the three mentioned tasks have constant-time computational costs (i.e. independent of the environment size). This is possible thanks to the estimated state structure, defined in (3). For a detailed proof, the reader is referred to [3].

On the other hand, the ESDF suffers from a strong limitation about the map precision. To be more precise, it suffers from the same limitations of every Gaussian-Filter-based solution to SLAM. The crucial problem is that a Gaussian-filter generally is a linear estimator. Unfortunately, the SLAM is a strong non linear problem, i.e. the robot motion function $f$ in (1) and the observation function $h$ in (2) are strongly non linear. This leads to the use of the linear approximation of both the robot kinematics and observations. In this way the estimation accuracy is obviously made worse.

## III. Improving ESDF by using a non linear Optimizer

The goal of this section is to introduce new changes in the ESDF in order to improve its performance when the non linearities are not negligible. The basic idea to achieve this objective is to combine the ESDF with a non linear optimizer, as suggested by the Graph-based approach. In order to do this it is more convenient to work with relative coordinates.

In the following we will first outline a key well known property of the EIF, then we will describe how to use the relative coordinates in the ESDF and finally how to combine the ESDF with a suitable non linear optimizer.

### A. Estimation process with the EIF

Let us refer to the equations (1) and (2) and let us focus on the observation update step of the EIF. The integration of the exteroceptive measurement $z_k$ is obtained by implementing the following equations:

$$\Lambda_k = \overline{\Lambda}_k + H_k^T R_k^{-1} H_k \tag{4}$$

$$\eta_k = \overline{\eta}_k + H_k^T R_k^{-1}\left(z_k - h\left(\overline{\mu}_k\right) + H_k \overline{\mu}_k\right) \tag{5}$$

where $R_k$ is the covariance matrix of the exteroceptive measurement error, $H_k$ is the Jacobian of the observation function $h$ computed in $\overline{\mu}_k$. The overline indicates the values of the quantities before the integration of the considered observation. We introduce the two following assumptions.

**Assumption 1 (Sparse Observation)** *The observation function $h$ only depends on $q$ components of $X_k$ and $q$ is independent of the size of $X_k$.*

**Assumption 2 (Easy State Recovery)** *It is possible to recover the estimated state $X_k$ (i.e. obtain $\mu_k$) from the information quantities $(\eta_k, \Lambda_k)$ with a complexity independent of the size of $X_k$.*

Under the previous assumptions we obtain the following property characterizing the complexity of the observation update.

**Property 1 (Observation Update Complexity)** *Under the assumptions 1 and 2 the observation update defined by the equations (4) and (5) can be computed with a complexity independent of the size of $X_k$, i.e. the observation update has a constant-time cost.*

**Proof:** if the observation function $h$ depends on $q$ elements of $X_k$, at any time step $k$, the integration of the information from the corresponding measurement requires to update only $q$ entries of $\eta_k$ and $q^2$ entries of $\Lambda_k$ (actually even less $\left(\frac{q(q+1)}{2}\right)$ because of the symmetry of $\Lambda_k$). Furthermore, the overall complexity is proportional to $q^2$. In the assumption 1 we suppose that $q$ is independent of the size of $X_k$. Therefore, if we suppose that the mean value $\overline{\mu}_k$ is available (assumption 2) the cost to implement the equations (4) and (5) is independent of the size of $X_k$.$\square$

In this paper we suppose that the state recovery problem is solved, i.e. we suppose that the assumption 2 is always satisfied. In [3] it is shown that it is possible to recover the mean value in a constant-time but its value will be approximated (see [3] for more details). Moreover, at any time, the robot typically makes a limited number, $q$, of relative observations to individual landmarks, i.e. a limited number of elements of the state $X_k$. This means that the assumption 1 is satisfied.

From property 1 we obtain that the ESDF observation update task has a constant-time computational cost.

### B. Using Relative Coordinates in ESDF

In this subsection we describe how to use the relative coordinates in the ESDF framework. In particular, we define the new coordinates to represent the same quantities estimated by ESDF, i.e. the robot poses and the landmark locations.

Before introducing the new coordinates, we define the structure of the new estimated state as it follows:

$$D_k^T = \left( D_k^{R^T} D_k^{L^T} \right) \tag{6}$$

where $D_k^R$ contains all the stored robot poses, and $D_k^L$ contains all the landmark locations.

*1) Robot Pose Coordinates:* Instead of defining the robot poses in a common global reference frame, each robot pose is defined in the frame of the robot at the previous time step. Let us indicate with $d_k^R$ the robot pose at the time step $k$ in the reference of the robot at the time step $k-1$, i.e. $x_k = x_{k-1} \oplus d_k^R$, where $\oplus$ is the composition operator between two robot poses. Therefore, the portion $D_k^R$ of the estimated state has the following structure:

$$D_k^{R^T} = (d_k^{R^T} d_{k-1}^{R}{}^T \cdots d_1^{R^T}). \tag{7}$$

Now, let us focus on the robot motion function $f$, defined in (1). It describes the relation between the current robot pose $x_k$ with the old robot pose $x_{k-1}$ and the proprioceptive measurement $u_k$, which is available at each time step. We can generally express this relation in the following way:

$$x_k = f(x_{k-1}, u_k + w_k) = x_{k-1} \oplus (u_k + w_k). \tag{8}$$

We are assuming that the proprioceptive measurements contain the necessary information to provide the shift and the rotation of the robot occurred at every step. This is for instance the case of the wheel encoders.

From the definition of the new coordinates of the robot pose $d_k^R$ and the equation (8) it follows that:

$$u_k = d_k^R + w_k. \tag{9}$$

The expression in (9) allows us to consider $u_k$ as a measurement of the estimated state. The idea is that the proprioceptive measurement can be considered as an observation of the estimated state: $u_k = \tilde{h}(D_k) + w_k$. If this is possible the proprioceptive measurement information can be integrated via the observation update defined in (4) and (5), applied to the measurement $u_k$. Furthermore, in our special case, the measurement function defined in (9) satisfies the hypothesis of sparse observation (assumption 1). This means that we can integrate the considered information with a constant-time computational cost.

*2) Landmark Coordinates:* Instead of defining the coordinates of each landmark in a global and unique reference frame, the new state defines a given landmark by its coordinates in the frame of the robot pose where it was observed for the first time. Let us indicate with $d_k^{L_j}$ the coordinates of the landmark $j$ in the reference attached to the robot pose at the time step $k$, i.e. we suppose that the landmark $j$ is observed at the time step $k$ for the first time. When the robot, at a given time step $l > k$ observes again this landmark, the relative measurement can be expressed by the following expression:

$$z_l = h(x_l, m_j) \tag{10}$$

where $m_j$ represents the coordinates of the landmark $j$ in the global frame. Since we are considering a relative measurement, the inputs of the function $h$ in (10) can be expressed in any reference frame (provided that it is the same for both inputs). By choosing the frame attached to the robot pose $x_k$ we have:

$$z_l = h(d_{k+1}^R \oplus \cdots \oplus d_l^R, d_k^{L_j}). \tag{11}$$

With the exception of the loop closure, the function $h$ depends on a number of elements of the estimated state which is independent of the size of the environment. To this regard, a loop closure event is defined as follows.

**Definition 1 (Loop Closure)** *The loop closure is the re-observation of a landmark after a while large enough to have at least one of the two following conditions:*

- *$l-k$ i.e. the number of elements $d_{k+q}^R$ ($q = 1, \ldots, l-k$) in (11), is large enough to make the execution of the observation update not possible in real time;*
- *the linearization of (11) makes the estimation process inconsistent.*

In order to detect the previous two conditions we propose the following criteria.

Regarding the first condition, we simply set a threshold on the computational time. As soon as the time required by the information filter to integrate a landmark re-observation exceeds this threshold, we consider the re-observation a loop closure.

Regarding the second condition, we propose the following criterion. Since we base on local relative coordinates we expect the innovation norm $\parallel z_k - h(\overline{\mu}_k) \parallel$ will be bounded by a given threshold. Once defined $\sigma^2$ as the max eigenvalue of the innovation covariance matrix, a possible threshold value can be $2\sigma$. Indeed, in a linear estimation process we know that the mentioned norm is bounded by $2\sigma$ with 0.95 likelihood. On the other hand, the non linearities could lead the innovation norm to overtake this threshold. When the norm is really bounded we are sure about the estimation consistency. On the contrary, when the innovation norm overtakes the threshold, we cannot say the same. Thus, this overtake event can be considered as a loop closure. Unfortunately, since we base on the information filter, the innovation covariance matrix is not available. However, basing on the measurements covariance matrices, we can build an approximated innovation covariance matrix whose norm is larger than that of the real one. In this way, we have a consistent threshold since it is larger than the theoretical one.

When a loop is closed, new coordinates corresponding to the re-observed landmark are introduced in the state. They are the coordinates of the landmark in the frame of the robot pose where the landmark is re-observed. Thus,

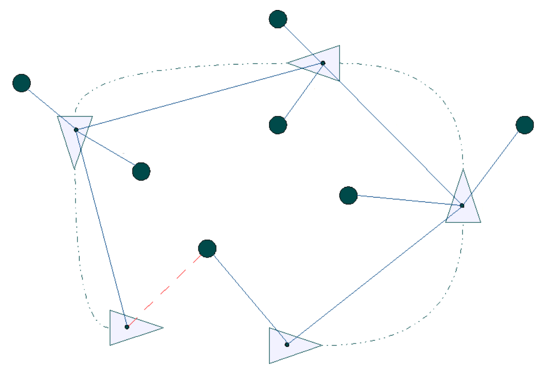

Fig. 1. The loop closure information. The blue discs represent the landmarks, the triangles represent the robot poses, the edges (blue and red dashed) represent the relative coordinates stored in the estimated state.

in this approach there are landmarks whose configuration is defined in more than one frame. This means that there are geometrical dependencies among state elements. These geometrical dependencies contain the information gained at the loop closure. We can say that by adding redundant coordinates we just freeze the loop closure information in these geometrical dependencies. This allows us to maintain the estimation process of the relative coordinates consistent and totally unaffected by the system non linearities. The exploitation of the information at the loop closure, namely of the previous geometrical dependencies, will be performed separately by a suitable non linear optimizer. We point out that this optimization can be computed only once, even if more than one loop closure event occurs.

### C. *Combining ESDF with a non linear Optimizer*

The basic idea consists in introducing a cost function. Such a cost function must carry the loop closure information, which is kept by the geometrical dependencies among the estimated state elements. Hence, it must be based on this geometrical information.

In order to simplify the notation, let us indicate the estimated state $D_k$ with $r$, the corresponding mean value with $\hat{r}$ and the information vector and matrix with $\eta$ and $\Lambda$, respectively. Furthermore, we indicate with $P$ the estimate error covariance matrix (i.e. inverse of $\Lambda$). We remark that both $\eta$ and $\Lambda$ are provided by our ESDF modification algorithm.

Let us focus on the example represented in figure 1. When the robot re-observes a landmark a loop is closed. The blue edges and the red dashed one represent all the relative quantities carried by the estimated vector $r$. The quantity represented by the red dashed edge can be expressed as a function of some of the other quantities, i.e. there are geometrical dependencies among state elements. In order to exploit this information, we introduce a new state containing only independent quantities. Possible choices are:

- the independent relative coordinates (e.g. the ones represented by the blue edges in figure 1);
- the global coordinates of both robot poses and landmarks in a common frame.

Let us indicate this state with $\tau$. As said, the quantities in $r$ can be expressed as a function of the components of $\tau$. Let us indicate this function with $\psi(\tau)$ (i.e. $r = \psi(\tau)$).

Our goal is to evaluate $\tau$ starting from $\Lambda$ and $\eta$. Let us indicate the best evaluation of $\tau$ with $\tau_{best}$.

$\tau_{best}$ minimizes the following cost function:

$$c(\tau) = (\hat{r} - \psi(\tau))^T \; P^{-1} \; (\hat{r} - \psi(\tau)) \tag{12}$$

namely $\tau_{best} = argmin_{\tau} c(\tau)$.

By expanding the expression of $c(\tau)$ and dropping the part independent of $\tau$, we obtain:

$$c(\tau) = \psi(\tau)^T \Lambda \; \psi(\tau) - 2\psi(\tau)^T \eta. \tag{13}$$

This last expression is very important since it shows that the computation of the cost function is based on the information quantities $\eta$ and $\Lambda$, namely it does not require to invert $\Lambda$.

Our method can now be completed by optimizing the cost function in (13) through a suitable optimization method. Literature provides lots of methods able to find a local minimum (or maximum) for a non linear function. We decided to use the well known *quasi-Newton*.

In order to use an optimization method we need to provide the cost function (13) computed for a given value of $\tau$ and the corresponding gradient. To do this, we must exactly define the meaning of the $\tau$ components and find the relation expressed by the function $\psi(\tau)$.

For our simulation, whose results can be found in the next section, we defined $\tau$ as the global coordinates of both the robot poses and the landmarks in a common frame. Therefore, the function $\psi(\tau)$ we obtained is made by inverse compositions which return the relative coordinates, given the absolute coordinates in $\tau$. Moreover, we observed that such a function is linear on the robot and landmarks location components of $\tau$, and non linear on the robot orientation components. This makes the cost function in (13) quadratic for the first mentioned portion of $\tau$ and non linear for the second portion. Thanks to this particular property, we minimized on the first portion through a suitable algebraical method (i.e. by solving a system of linear equations). Then, we minimized on the second portion through the non linear optimizer. This algebraical manipulation reduced the dimension of the space in which the optimization algorithm had to move, making the optimization significantly faster.

The described optimization method is obviously able to only find a local minimum of the cost function (13). Actually, in our framework it would be better to find a global minimum. Indeed, there exist situations in which a local minimum does not represent the best solution to the mapping problem. A deterministic method able to find a global minimum does not exist. Nevertheless, there are method, such as the *Stochastic Gradient Descent* (from now on SGD), which attempt to find the most "popular" minimum (i.e. the likelihood for the reached minimum to not be global is made low as much as possible). In the Graph-based approach some improvements of the SGD are suggested to solve the general SLAM minimization problem.

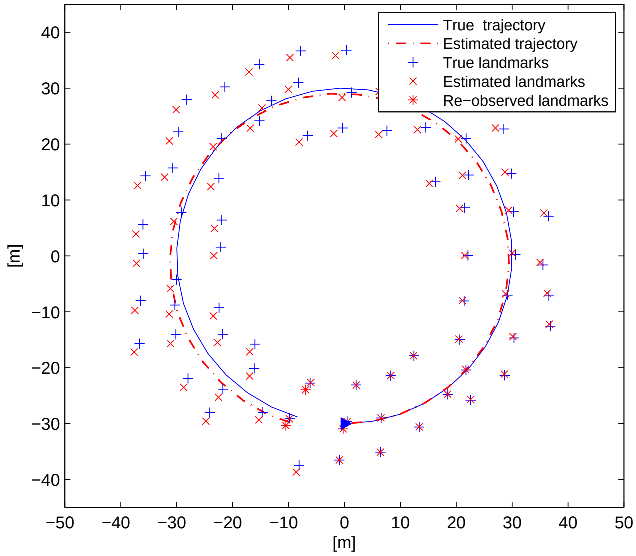


Fig. 2. Simulation 1: estimation results of our method before the loop closure.

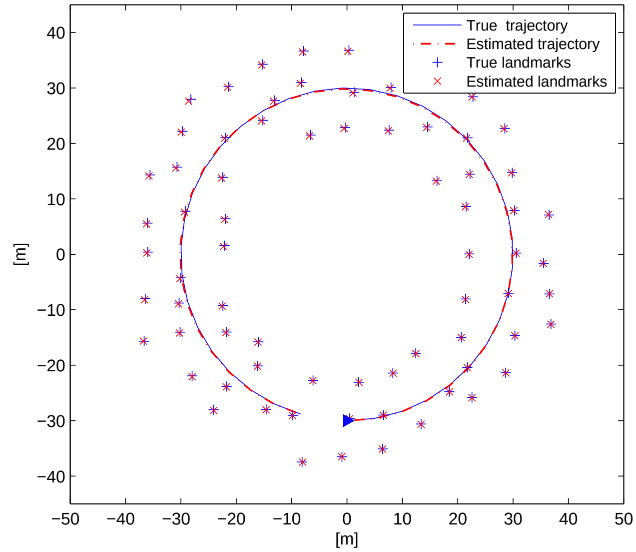


Fig. 4. Simulation 1: estimation results of our method after the loop closure.

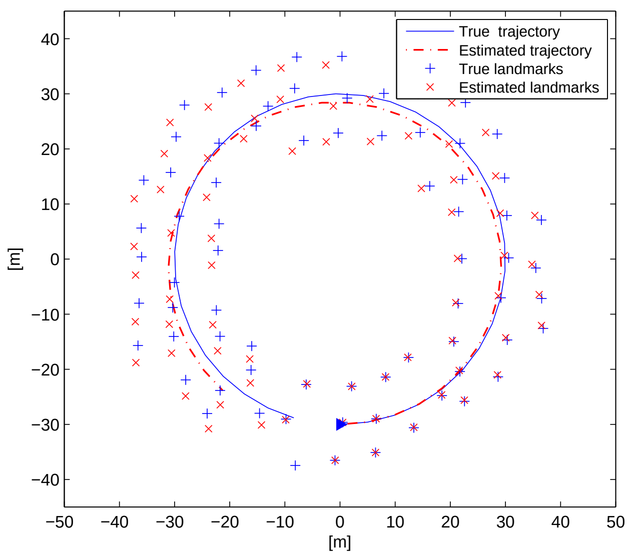


Fig. 3. Simulation 1: ESDF estimation results before the loop closure.

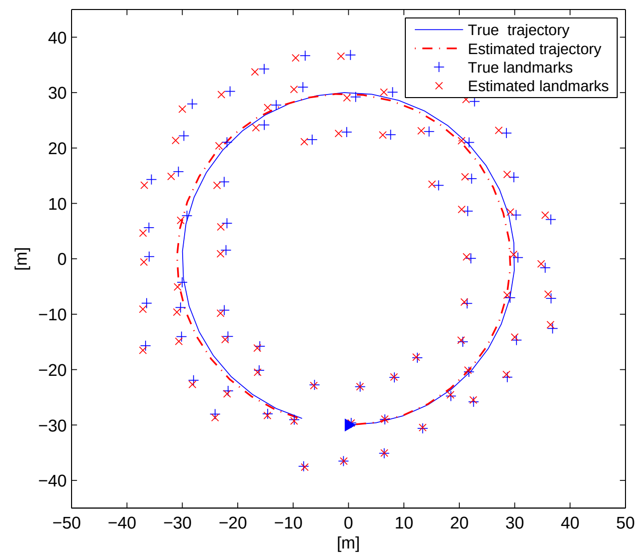


Fig. 5. Simulation 1: ESDF estimation results after the loop closure.

As said in section I, in this paper we only take the basic idea of the Graph-based approach. Thus, we restrict ourself to find local minima. As the results in section IV show, this could be enough to find a very accurate map in our framework since the optimization process starts from the pre-elaborated data returned by the filtering process. On the contrary, in the Graph-based approach the optimization directly starts from the measurements.

However, our future works will be developed in order to apply the SGD in our framework. In this way, we will give a more general solution to our problem.

## IV. Results

In order to evaluate the performances of the proposed method, we performed many simulations. We considered a conventional scenario defined by a few parameters which regard the robot perception and the environment properties. In the following, we will first define this scenario clearly pointing out the meaning of some simulation parameters. Then, we will provide the results of a few simulations in which we analyze the behavior of our method. Furthermore, we will compare the performances of our approach with those of the ESDF one.

All our simulations are implemented in Matlab and tested on a computer with 1 Intel Pentium CPU M 1.70GHz, 512MB of memory.

### A. Simulated scenario

In our simulations we consider a two-wheels robot moving in a $110m$x$110m$ rectangular area in which many point landmarks are randomly distributed. Let us indicate the average landmark density with $\rho_L$, the robot average speed

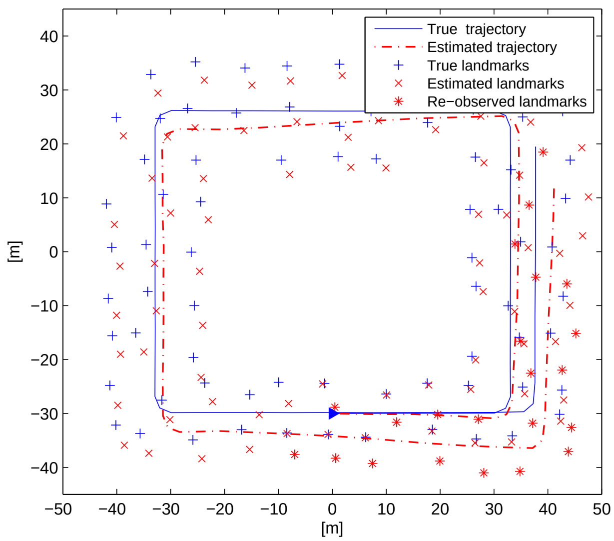


Fig. 6. Simulation 2: estimation results of our method before the loop closure.

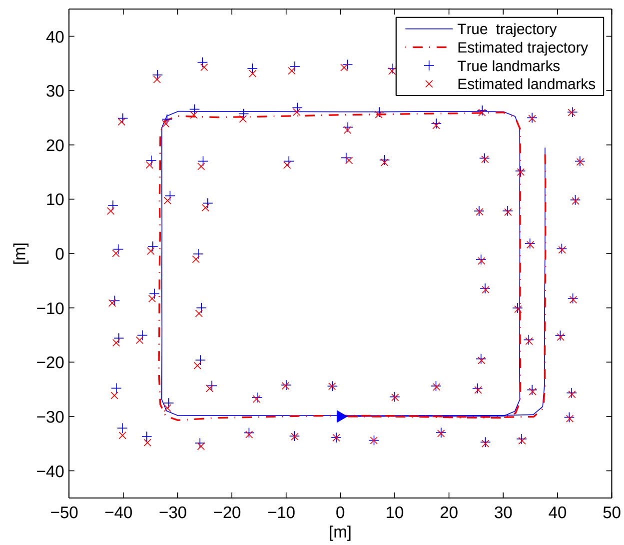


Fig. 8. Simulation 2: estimation results of our method after the loop closure.

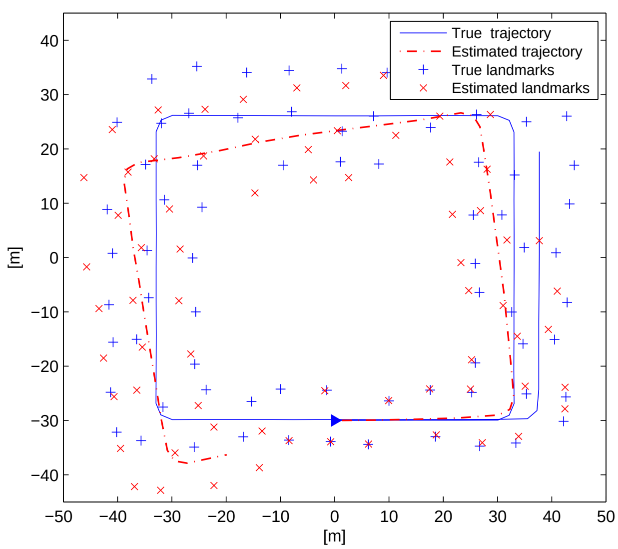


Fig. 7. Simulation 2: ESDF estimation results before the loop closure.

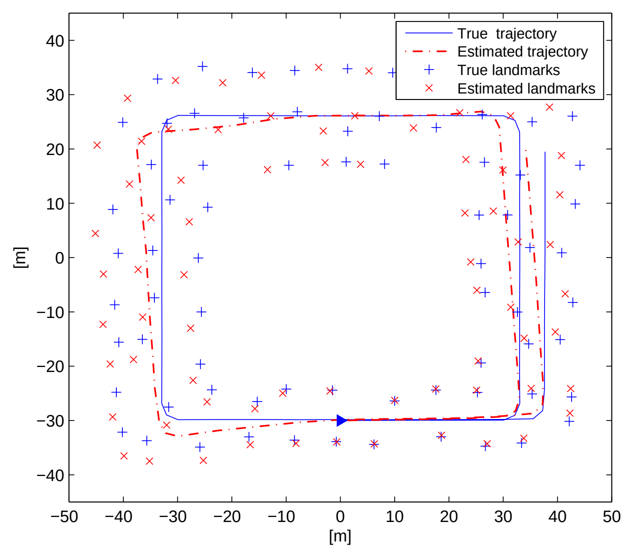


Fig. 9. Simulation 2: ESDF estimation results after the loop closure.

with $v$, and the distance traveled by the robot with $d$. The data associations ere supposed to be given. We consider a robot equipped with wheel encoders which provide the proprioceptive measurements. We base on the Chong-Kleeman ([2]) error model. According to this model, the translation of the right/left wheel as estimated by the odometry sensors is generated as a Gaussian random quantity
satisfying the following relations:

$$\delta\rho^{R/L} = \delta\rho^{aR/L}\delta_{R/L} + \nu^{R/L}, \qquad (14)$$
$$\nu^{R/L} \sim N(0, K|\delta\rho^{aR/L}|).$$

In other words, both $\delta\rho^R$ and $\delta\rho^L$ are assumed Gaussian random variables, whose mean values are given by the actual values (respectively, $\delta\rho^{aR}$ and $\delta\rho^{aL}$), and whose variance also increases linearly with the travelled distance. In our simulation we set $K = 0.00001m$, which corresponds to an indoor environment [7]. Finally, the frequency is $100Hz$.

The simulated exteroceptive sensor provides bearings and ranges of the landmarks whose distance does not exceed $12m$. Furthermore, the sensor angle of view is $360deg$. Both the bearings and the distances are generated as Gaussian quantities with variances equal to $\sigma_R^2$ and $\sigma_B^2$, respectively. The frequency is $0.2Hz$.

## B. Performance Evaluation

Figs 2-5 show the results provided by a given simulation in which the robot closes a loop in counter clockwise direction. Let us point out that the loop closure does not consist of the trajectory closure but it consists of the re-observation of landmarks located close to the starting point.

We implemented both our method and the ESDF one. As said in section III, the minimization is carried out through a *quasi-Newton* method. We set $\sigma_B = 1deg$, $\sigma_R = 1cm$,

$\rho_L = 0.02m^{-2}$, $v = 1ms^{-1}$, $d = 180m$. The simulation time is $T_s = 180s$.

Figs 2 and 3 show the results obtained before the loop closure. In each figure, we represent the true robot trajectory and the true landmark locations (ground truth) by a solid blue line and cross blue markers, respectively. Moreover, the estimated trajectory and landmark locations are represented by a dash-dotted red line and red x-markers, respectively.

In order to provide quantitative results, we consider the error on the estimated map by computing for all the landmarks the distance between the estimated location and the corresponding true location. Then, the mean value on all the landmarks is taken. We refer to this mean value as the *map error* ($E_m^{bl}$ before the loop closure and $E_m^{al}$ after the loop closure).

The behavior of our estimation process and that of the ESDF one are very similar. However, the map errors are $E_m^{bl} = 1.30m$ for our method and $E_m^{bl} = 2.02m$ for the ESDF. Therefore, our method shows a better behavior also before the loop closure.

Figs 4 and 5 show the results after the loop closure. The correction we obtained through the non linear optimizer clearly outperforms the one computed by the ESDF method. This is confirmed by the map errors: $E_m^{al} = 0.15m$ for our method and $E_m^{al} = 1.01m$ for the ESDF.

The total computation time needed for the estimation process is $Tc = 16.20s$ for our method ($5.45s$ for the filtering process and $10.75s$ for the optimization) and $Tc = 39.67s$ for the ESDF.

Figs 6-9 show the results provided by a second simulation in which the robot closes a loop in counter clockwise direction and than goes on re-traversing a region for a long time. The parameters of this simulation are: $\sigma_B = 1deg$, $\sigma_R = 1cm$, $\rho_L = 0.02m^{-2}$, $v = 1.2ms^{-1}$, $d = 325m$. The simulation time is $T_s = 270s$.

Figs 6 and 7 show the two methods before exploiting the loop closure information. Concerning our method, as said in section III, a loop closure does not necessary activate the optimization . Indeed, in this case the estimation process goes on considering the re-observed landmarks as new landmarks. In figure 6 these phantom landmarks are represented by star red markers. As the figures clearly show, our estimation process outperforms the ESDF one. This is confirmed by the map errors: $E_m^{bl} = 3.17m$ for our method and $E_m^{bl} = 3.91m$ for the ESDF.

Figure 8 shows the results obtained through the non linear optimizer which is activated only once, after a long time from the first loop closure. Moreover, figure 9 shows the results of the correction computed by the ESDF technique after the loop closure. The comparison of these two last figures clearly shows the success of our hybrid approach in improving the ESDF performances. This is confirmed by the map errors: $E_m^{al} = 0.38m$ for our method and $E_m^{al} = 0.58m$ for the ESDF.

The total computation time needed for the estimation process is $Tc = 46.36s$ for our method ($13.67s$ for the filtering process and $32.70s$ for the optimization) and $Tc = 91.44s$ for the ESDF.

## V. Conclusions

In this paper we considered the SLAM problem. Currently, the community has to solve two contrasting problems, which are often faced with a trade-off: the map precision and the computational requirement for real-time/real-world implementations. In order to propose a SLAM solution able to face with both the problems, we decided to combine two different approaches: the Filter-based approach and the Graph-based approach ([8],[4]).

In particular, we considered ESDF technique ([3]) which makes possible a real-time/real-world implementation for any kind of environment. The only drawback of this technique is the use of the linear approximation which could become not consistent when the environment is large enough.

On the contrary, in the Graph-based approach, SLAM is solved as a non linear optimization problem. This leads to more accurate results.

Therefore, we proposed a method able to combine a suitable modification of the ESDF with a non linear optimizer. This solution allows us to use the modified ESDF when the non linearities are negligible and to switch to the optimizer when the non linearities are not negligible.

The results provided in section IV show how our hybrid approach succeeds in outperforming the ESDF in terms of map precision.

## References


[1] J. A. Castellanos, J. Neira, and J. D. Tardos. Limits to consistency of the EKF-based SLAM. In *Intelligent Autonomous Vehicle (IAV 2004)*, Lisbon, Portugal, July 2004.

[2] K. S. Chong and L. Kleeman. Accurate odometry and error modelling for a mobile robot. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA97)*, Albuquerque, New Mexico, USA, April 1997.

[3] R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transaction on Robotics*, 22(6), December 2006.

[4] G. Grisetti, C. Stachniss, S. Grzonka, and W. Bugard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

[5] S. Julier and J. K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *International Conference on Robotics and Automation (ICRA01)*, Seoul, Korea, 2001.

[6] A. Martinelli and R. Siegwart. Exploiting the information at the loop closure in SLAM. In *International Conference on Robotics and Automation (ICRA 2007)*, Rome, Italy, April 2007.

[7] A. Martinelli, N. Tomatis, and R. Siegwart. Simultaneous localization and odometry self calibration for mobile robot. *Autonomous Robot*, 22:75–85, 2007.

[8] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *International Conference on Robotics and Automation (ICRA06)*, Orlando, Florida,USA, 2006.