9

Cooperative Localization and SLAM Based on the Extended Information Filter

Francesco Conte[1], Andrea Cristofaro[2],
Alessandro Renzaglia[2] and Agostino Martinelli[2]
[1]*Università degli Studi dell'Aquila*
[2]*INRIA Rhône-Alpes*
[1]*Italy*
[2]*France*

1. Introduction

Simultaneous Localization and Mapping (SLAM) requires a mobile robot to autonomously explore the environment with its on-board sensors, gain knowledge about it, interpret the scene, build an appropriate map and localize itself relative to this map. Many approaches have been proposed both in the framework of metric and topological navigation. A very successful metric method is the stochastic map (26) where early experiments (4) (16) have shown the quality of fully metric SLAM.
Currently, the SLAM has two contrasting problems to be solved, which are often faced with a trade-off:

– The map precision;

– The computational requirement for real-time/real-world implementation

Dissanayake et al. (5), proved the convergence of an algorithm based on the Kalman filter theoretically. However, the proof is based on the strong hypothesis of a linear observation. Julier and Uhlmann (13) and Castellanos et al. (2) proved that the conventional EKF based $SLAM$ (from now on $EKF-SLAM$) yields an inconsistent map (in particular, in (13) was shown that this happens even for the special case of a stationary vehicle with no process noise). The map inconsistency arises from the linearization introduced by the Extended Kalman Filter (EKF) as clearly pointed out by Castellanos et al. (2). 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 vehicle location will drift (if no loop is closed). This is a consequence of the fact that the absolute location is derived from a composition of many relative measurements. Therefore, when the drift is large enough, the linearization is not a possible approximation. Additionally, even by solving SLAM with an EKF (i.e. by adopting a linearization) the computational complexity becomes prohibitive since it grows up squarely in the features number.
In 2006 Eustice *et al,* (see (6)) introduced an approach called *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. 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 (22) and Grisetti (9) 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 the first part of this chapter, we illustrate 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 (18) in the context of the Relative Map Approach to solve SLAM.
In the second part of this chapter we will focus our attention on the cooperative case, i.e. when the estimation process is performed simultaneously by a team of agents. In particular, we consider the case of flying vehicles. In recent years, flying robotics has received significant attention from the robotics community. The ability to fly allows easily avoiding obstacles and quickly having an excellent birds eye view. These navigation facilities make flying robots the ideal platform to solve many tasks like exploration, mapping, reconnaissance for search and rescue, environment monitoring, security surveillance, inspection etc. In the framework of flying robotics, micro aerial vehicles (MAV) have a further advantage. Due to the small size they can also be used in narrow out- and indoor environment and they represent only a limited risk for the environment and people living in it. One of the main prerequisite for the successful accomplishment of many tasks is a precise vehicle localization. Since micro aerial vehicles are equipped with low computational capabilities an efficient solution must be able to distribute the computation among all the agents in order to exploit the computational resources of the entire team. Distributing the computation has also another key advantage. It allows us to make the solution robust with respect to failures. On the other hand, distributing the computation must also account the limited communication capabilities.
The cooperative localization problem has been faced by many authors so far. Fox and collaborators (7) introduced a probabilistic approach based on Markov localization. Their approach has been validated through real experiments showing a drastic improvement in localization speed and accuracy when compared to conventional single robot localization. Other approaches take advantage of relative observations for multi-robot localization (8; 10; 15; 23; 24; 27). In (10) a method based on a combination of maximum likelihood estimation and numerical optimization was introduced. This method allows to reduce the error in the robot localization by using the information coming from relative observations among the robots in the team. In (24), a distributed multi robot localization strategy was introduced. This strategy is based on an Extended Kalman Filter to fuse proprioceptive and exteroceptive sensor data. In (17), the same approach was adapted in order to deal with any kind of relative

observations among the robots. In (24), it was shown that the equations can be written in a decentralized form, allowing the decomposition into a number of smaller communicating filters. However, the distributed structure of the filter only regards the integration of the proprioceptive data (i.e. the so called prediction phase). As soon as an observation between two robots occurs, communication between each member of the team and a single processor (which could be embedded in a member of the team) is required. The same communication skill is required when even an exteroceptive measurements which only regards a single robot occurs (e.g. a GPS measurement). Furthermore, the computation required to integrate the information coming from this observation is entirely performed by a single processor with a computational complexity which scales quadratically with the number of robots. Obviously, the centralized structure of the solution in dealing with exteroceptive observations becomes a serious inconvenient when the communication and processing capabilities do not allow to integrate the information contained in the exteroceptive data in real time. In particular, this happens as soon as the number of robots is large, even if each robot performs very few exteroceptive observations. In (20) this problem was considered. However, the structure of the filter was maintained the same as in (24) (namely centralized in dealing with exteroceptive data). Each robot was supposed to be equipped with several sensors and the optimal sensing frequencies were analytically derived by maximizing the final localization accuracy. The limit of this approach is that as the number of robots increases, the sensing frequencies reduce. In other words, by performing the estimation process in a centralized fashion it is necessary to reduce the number of observations to be processed as the number of robots increases. Hence, distibuting the entire estimation process can provide a great improvement.

The information filter is very appealing in this framework since the integration of the exteroceptive data is very simple and could be easily distributed. On the other hand, the equations which characterize the prediction step are much more complex and their distributed implementation seems to be forbidden. This is a serious inconvenient since the proprioceptive data run at a very high frequency.

Eustice et al. (6) and Caballero et al. (1) have recently shown that by using a delayed state also the prediction step has some nice properties. In particular, in (6) a solution to the SLAM problem by using an Extended Information Filter (EIF) to estimate a delayed state has been proposed. In (1) the tracking problem has been considered.

In the second part of this chapter we present the problem of cooperative localization in $3D$ when the MAVs are equipped with inertial sensors and exteroceptive sensors (e.g. range sensors and GPS). We adopt a delayed state and we perform its estimation by using an Extended Information Filter. We introduce a simple trick which allows us to mathematically express the quantities measured by the IMU (Inertial Measurement Unit) as a function of the delayed state (i.e. the state to be estimated). In other words, by using this trick, the link between sensor-state for the IMU (which are typically proprioceptive sensors) has the same mathematical expression of the one which characterizes an exteroceptive observation. This allows us to use the equations of the integration of the exteroceptive data also to integrate the IMU data. In this way the equations of the EIF prediction step are never used and the overall estimation process can be easily distributed.

When dealing with a $3D$ environment, another important issue arises. The orientation of a MAV which moves in $3D$ is provided by 3 parameters. On the other hand, the MAV dynamics become very easy by adopting quaternions. However, this parameterization is redundant. This means that part of the information is frozen in a geometrical constraint. Without using this constraint part of the information is not exploited and the overall precision gets worse.

To this regard a new filter, the projection filter, has been introduced (21); this filter permits us to consider the geometrical constraint (expressing that the quaternion must be unitary) as an ideal observation.

2. A brief overview on extended information filter

Consider an arbitrary system driven by the discrete equations

$$x_i = f(x_{i-1}, u_i)$$

$$z_i = h(x_i)$$

Let us denote with Σ and ξ the information matrix and the information vector respectively; we recall that information matrix and information vector are related to covariance matrix P and mean value μ as follows

$$\Sigma = P^{-1}, \quad \xi = P^{-1}\mu.$$

2.1 Estimation with EIF: integration of exteroceptive data

Let R be the covariance matrix characterizing the measurement error for an exteroceptive sensor. The update equations at the time step i are (see (28)):

$$\Sigma_i = \overline{\Sigma}_i + \Sigma_{obs}, \qquad \Sigma_{obs} = H_i^T R^{-1} H_i, \tag{1}$$

$$\xi_i = \overline{\xi}_i + \xi_{obs}, \qquad \xi_{obs} = H_i^T R^{-1} \left[z_i - h(\overline{\mu}_i) + H_i \overline{\mu}_i \right], \tag{2}$$

where $\overline{\Sigma}_i, \overline{\xi}_i$ are the predicted information matrix and information vector, $\overline{\mu}_i = \overline{\Sigma}_i^{-1} \overline{\xi}_i$ is the predicted mean value and H_i is the Jacobian of the observation function $h(\cdot)$ evaluated at $\overline{\mu}_i$.

2.2 Estimation with EIF: integration of proprioceptive data

Denoting by Q a noise term affecting the system dynamics, the prediction steps are given by

$$\overline{\Sigma}_i = \left[F_i \Sigma_{i-1}^{-1} F_i^T + Q \right]^{-1}, \tag{3}$$

$$\overline{\xi}_i = \overline{\Sigma}_i F_i \Sigma_{i-1}^{-1} \xi_{i-1}, \tag{4}$$

where F_i is the Jacobian of the dynamics $f(\cdot,\cdot)$ evaluated at the estimated mean value (μ_{i-1}, u_i), where $\mu_{i-1} = \Sigma_{i-1}^{(-1)} \xi_{i-1}$.

2.3 EIF and delayed-states

In a multi robot scenario Σ and ξ characterize the probability distribution of several robots; in (1) it is shown that delayed-states allow us to distribute the estimation process over the entire network. In particular the authors explain how to recover the global belief from the local belief of each network node and remark that the same operation with standard (non delayed) states is not possible at all.

Definition 1 *A delayed-state is a dynamic vector X whose entries at time step i are the current robot coordinates x_i together with all the past poses $x_0, x_1, ..., x_{i-1}$.*

For example, a delayed-state for a 2D robot having coordinates assigned by $x_i = (r_{x,i}, r_{y,i}, \theta_i)$ is given by the vector

$$X_i = (r_{x,0}, r_{y,0}, \theta_0, r_{x,1}, r_{y,1}, \theta_1, ..., r_{x,i}, r_{y,i}, \theta_i).$$

As already mentioned, a distributed algorithm for the implementation of update equations (1)-(2) can be designed (see (1)). The structure of such equation is very simple as the update consists only in summing the new information from the exteroceptive sensors to the predicted values. On the other hand, the prediction equations (3)-(4) are more complicated and they cannot be easily distributed. Nevertheless we will show that, once a delayed-state is considered, data obtained from proprioceptive sensors can be integrated using only the update equations (1)-(2).

3. SLAM problem for a single 2D robot

3.1 Advantages and drawbacks in the ESDF-approach

In 2006 Eustice *et al.* (6) 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 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:

$$x_i = f(x_{i-1}, u_i + w_i) \tag{5}$$

$$z_i = h(x_i, m) + v_i \tag{6}$$

where x_i is the robot pose at the time step i, u_i is the control input (proprioceptive measurement), z_i is the exteroceptive measurement available at the time step i, m is the environment map, f is the robot motion function, h is the observation function, and w_i and v_i 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 i the current estimated vector is:

$$X_i^T = \begin{pmatrix} x_i^T & M^T \end{pmatrix} \tag{7}$$

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_i as the environment size.

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 μ_i 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 (7). For a detailed proof, the reader is referred to (6).

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 (5)

and the observation function h in (6) 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.

3.1.1 Estimation process with the EIF

The estimation process is performed using update and prediciton steps given by (1)-(2) and (3)-(4). We introduce the two following assumptions.

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

Assumption 2 (Easy State Recovery) *It is possible to recover the estimated state X_i (i.e. obtain μ_i) from the information quantities (ξ_i, Σ_i) with a complexity independent of the size of X_i.*

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 (1) and (2) can be computed with a complexity independent of the size of X_i, i.e. the observation update has a constant-time cost.*

Proof: if the observation function h depends on q elements of X_i, at any time step k, the integration of the information from the corresponding measurement requires to update only q entries of ξ_i and q^2 entries of Σ_i (actually even less $\left(\frac{q(q+1)}{2}\right)$ because of the symmetry of Σ_i). Furthermore, the overall complexity is proportional to q^2. In the assumption 1 we suppose that q is independent of the size of X_i. Therefore, if we suppose that the mean value $\overline{\mu}_i$ is available (assumption 2) the cost to implement the equations (1) and (2) is independent of the size of X_i. In the following we will suppose that the state recovery problem is solved, i.e. we suppose that the assumption 2 is always satisfied. In (6) it is shown that it is possible to recover the mean value in a constant-time but its value will be approximated (see (6) 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_i. 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.

3.2 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:

$$D_i^T = \left(D_i^{R\,T} D_i^{L\,T} \right) \tag{8}$$

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

3.2.1 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_i^R the robot pose at the time step k in the reference of the robot at the time step $k-1$, i.e. $x_i = x_{i-1} \oplus d_i^R$, where $\oplus$ is the composition operator between two robot poses. Therefore, the portion D_i^R of the estimated state has the following structure:

$$D_i^{R\,T} = (d_i^{R\,T}\, d_{i-1}^{R\;\;T} \cdots d_1^{R\,T}) \tag{9}$$

Now, let us focus on the robot motion function f, defined in (5). It describes the relation between the current robot pose x_i with the old robot pose x_{i-1} and the proprioceptive measurement u_i, which is available at each time step. We can generally express this relation in the following way:

$$x_i = f(x_{i-1}, u_i + w_i) = x_{i-1} \oplus (u_i + w_i) \tag{10}$$

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_i^R and the equation (10) it follows that:

$$u_i = d_i^R + w_i \tag{11}$$

The expression in (11) allows us to consider u_i as a measurement of the estimated state. The idea is that the proprioceptive measurement can be considered as an observation of the estimated state: $u_i = \tilde{h}(D_i) + w_i$ and hence the proprioceptive measurement information can be integrated via the observation update defined in (1) and (2), applied to the measurement u_i. Furthermore, in our special case, the measurement function defined in (11) satisfies the hypothesis of sparse observation (assumption 1). This means that we can integrate the considered information with a constant-time computational cost.

3.2.2 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 $d_i^{L_j}$ 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:

$$z_l = h(x_l, m_j) \tag{12}$$

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 (12) 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_i we have:

$$z_l = h(d_{i+1}^R \oplus \cdots \oplus d_l^R, d_i^{L_j}) \tag{13}$$

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 2 (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:*

- *$l - i$ i.e. the number of elements d^R_{i+q} ($q = 1, \ldots, l - i$) in (13), is large enough to make the execution of the observation update not possible in real time;*
- *the linearization of (13) 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 $\| z_i - h(\overline{\mu}_i) \|$ will be bounded by a given threshold. Once defined σ^2 as the max eigenvalue of the innovation covariance matrix, a possible threshold value can be 2σ. Indeed, in a linear estimation process we know that the mentioned norm is bounded by 2σ 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, 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.

3.3 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_i with r, the corresponding mean value with $\hat{r}$ and the information vector and matrix with ξ and Σ, respectively. Furthermore, we indicate with P the estimate error covariance matrix (i.e. inverse of Σ). We remark that both ξ and Σ 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

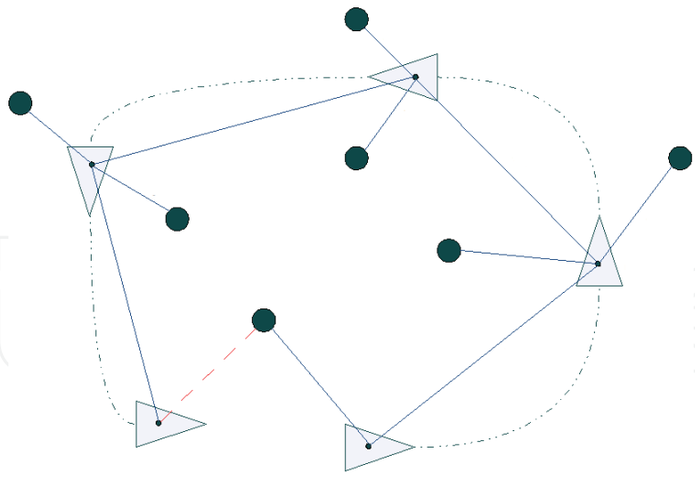


Fig. 1. The loop closure information. The blue discs represent the landmarks, the triangles represent the robot poses, the edges (blue and red dashed) represent the relative coordinates stored in the estimated state.

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 τ. As said, the quantities in r can be expressed as a function of the components of τ. Let us indicate this function with $\psi(\tau)$ (i.e. $r = \psi(\tau)$).
Our goal is to evaluate τ starting from Σ and ξ. Let us indicate the best evaluation of τ with τ_{best}. τ_{best} minimizes the following cost function:

$$c(\tau) = (\hat{r} - \psi(\tau))^T \; P^{-1} \; (\hat{r} - \psi(\tau)) \tag{14}$$

namely $\tau_{best} = argmin_\tau c(\tau)$.
By expanding the expression of $c(\tau)$ and dropping the part independent of τ, we obtain:

$$c(\tau) = \psi(\tau)^T \Sigma \, \psi(\tau) - 2\psi(\tau)^T \xi \tag{15}$$

This last expression is very important since it shows that the computation of the cost function is based on the information quantities ξ and Σ, namely it does not require to invert Σ.
Our method can now be completed by optimizing the cost function in (15) 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 (15) computed for a given value of τ and the corresponding gradient. To do this, we must exactly define the meaning of the τ components and find the relation expressed by the function $\psi(\tau)$.
For our simulation, whose results can be found in section 5, we defined τ 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 τ. Moreover, we observed that such a function

is linear on the robot and landmarks location components of τ, and non linear on the robot orientation components. This makes the cost function in (15) quadratic for the first mentioned portion of τ 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.

4. Localization problem for MAV systems

4.1 A single MAV system

We provide here a mathematical description of our system. We introduce a global frame, whose z-axis is the vertical one. Let us consider a MAV equipped with IMU proprioceptive sensors (an accelerometer and a gyroscope) as well as some suitable exteroceptive sensors (GPS, range sensors). In the following we assume that the IMU data are unbiased. From a practical point of view, unbiased data can be obtained by continuously calibrating the IMU sensors (see for instance (11)). The configuration of the MAV is described by a vector $(r,v,\theta) \in \mathbf{R}^9$ where $r = (r_x, r_y, r_z) \in \mathbf{R}^3$ is the position, $v = (v_x, v_y, v_z) \in \mathbf{R}^3$ is the speed and $\theta = (\theta_\mathrm{r}, \theta_\mathrm{p}, \theta_\mathrm{y}) \in \mathbf{R}^3$ assignes the MAV orientation: θ_r is the roll angle, θ_p is the pitch angle and θ_y is the yaw angle. We will adopt lower case letters to express a quantity in the global frame, while capital letters for the same quantity exptressed in the local frame (i.e. the one attached to the MAV).The system description can be simplified adopting a quaternions framework. We recall that the quaternions space $\mathbf{H}$ is the noncommutative set of elements

$$\mathbf{H} = \left\{ q_t + q_x i + q_y j + q_z k : \quad q_t, q_x, q_y, q_z \in \mathbf{R}, \quad i^2 = j^2 = k^2 = ijk = -1 \right\}.$$

For an arbitrary quaternion $q = q_t + q_x i + q_y j + q_z k$, we define the conjugate element $q^* = q_t - q_x i - q_y j - q_z k$ and the norm $||q|| = \sqrt{qq^*} = \sqrt{q^* q} = \sqrt{q_t^2 + q_x^2 + q_y^2 + q_z^2}$.
Denoting by A, Ω the accelerometer and the gyroscope values respectively and by a_g the gravity acceleration (i.e. $a_g = -(0,0,g)$ with $g \simeq 9.81 m/s^2$), the continuous-time dynamics of the MAV is given by the following system of ordinary differential equations

$$\dot{r} = v \tag{16}$$

$$\dot{v} = q \cdot A \cdot q^* + a_g \tag{17}$$

$$\dot{q} = \frac{1}{2} q \cdot \Omega \tag{18}$$

where r, v, Ω, A are purely imaginary quaternions, while q is a unitary quaternion. The following relations for roll, pitch and yaw angles $\theta_\mathrm{r}, \theta_\mathrm{p}, \theta_\mathrm{y}$ hold

$$\theta_\mathrm{r} = \frac{q_t q_x + q_y q_z}{1 - 2(q_x^2 + q_y^2)}$$

$$\theta_\mathrm{p} = q_t q_y - q_x q_z$$

$$\theta_\mathrm{y} = \frac{q_t q_z + q_y q_x}{1 - 2(q_y^2 + q_z^2)}.$$

During the exploration, the MAV performs measurements thanks to its exteroceptive sensors equipment; such measurements can be individual (i.e. GPS-based measurements) as well as relative to other MAVs poses or to the position of fixed landmarks. The general single MAV observation equation is given by

$$z = h(r, v, q) \tag{19}$$

where $h(\cdot,\cdot,\cdot)$ is a known function.
In the case the exteroceptive sensor is a GPS, the observation equation is very simple as it is linear

$$z_{GPS} = r. \tag{20}$$

4.1.1 Estimation with the EIF: the integration of the proprioceptive data

Let us introduce the delayed-state

$$X_i = (r_0, q_0, r_1, ..., r_i, q_i)$$

containing all MAV poses until the i-th time step. The discretization of the dynamics equations over a Δt time-step interval gives

$$r_{i+1} = r_i + v_i \Delta t \tag{21}$$

$$v_{i+1} = v_i + q_i \cdot \int_i^{i+\Delta t} A dt \cdot q_i^* + a_g \Delta t \tag{22}$$

$$q_{i+1} = q_i + \frac{1}{2} q_i \cdot \int_i^{i+\Delta t} \Omega dt \tag{23}$$

From Equation (21) we can get

$$v_i = (r_{i+1} - r_i)/\Delta t$$

and hence the following recursive formula holds

$$r_{i+1} = 2r_i - r_{i-1} + \Delta t \left(q_i \cdot \int_i^{i+\Delta t} A dt \cdot q_i^* + a_g \Delta t \right), \tag{24}$$

corresponding to a second order continuous-time evolution. The system dynamics can be written in terms of delayed-states as

$$X_{i+1} = \mathcal{F}(X_i),$$

where $\mathcal{F}$ is a suitable function obtained from (23)-(24). Setting

$$\tilde{A}_i = \int_i^{i+\Delta t} A dt \tag{25}$$

and

$$\tilde{\Omega}_i = \int_i^{i+\Delta t} \Omega dt, \tag{26}$$

the proprioceptive measurements can be regarded as delayed-state dependent functions:

$$\tilde{A}_i = h_A(r_{i-2}, r_{i-1}, r_i, q_i) = \frac{q_i^*(-a_g \Delta t^2 + r_i - 2r_{i-1} + r_{i-2}) q_i}{\Delta t}$$

$$\tilde{\Omega}_i = h_\Omega(q_{i-1}, q_i) = 2q_{i-1}^*(q_i - q_{i-1}).$$

In other words, $\tilde{A}_i$ and $\tilde{\Omega}_i$ are functions of the state X_i to be estimated; moreover, since we are considering the discrete dynamics given by (23)-(24), there is no need to include the MAV speed v into the state vector X_i.
Due to these considerations, we are allowed to integrate proprioceptive data using (1)-(2) instead of (3)-(4), with a consequent reduction of computational cost in the estimation algorithm.
For nonlinear measurements equation (2) involves the mean value and hence information matrix inversion is required; nevertheless in many situation, due to the sparsity of such matrix, a partial state recovery is sufficient in order to guarantee a good estimate (see (6)). Whole state recovering can be obtained using for example the Conjugate Gradients algorithm (see (25)) or the Givens rotations factorization (see (14)). We point out that at any update step, i.e. when a true exteroceptive measurement is performed, the size of the delayed-state vector X increases of $3 + 4 = 7$.

4.1.2 Projection filter: integration of ideal constraints

As mentioned in the introduction, the quaternion structure is redundant for the problem we are considering and this may lead to a loss of information. To avoid this problem we have assumed that the quaternion q is unitary. On the other hand, if the discrete dynamics (23) is considered, such property is no longer preserved. Anyway, we can take into account the norm invariance of q_i imposing an ideal constraint with a fake observation given by the function

$$h_0(q) = 1 - q_t^2 + q_x^2 + q_y^2 + q_z^2;$$

in other words, we can regard the norm constraint as the measurement

$$z_i = h_0(q_i) = 0.$$

Integration of such fake measurement can be performed with the projection filter (see (21)).

4.2 The cooperative case

We consider now the problem of cooperaive localization for a multi robot system.

4.2.1 The system

We consider now a fleet of $N > 1$ MAVs, each one having the characteristics described in Section 4.1. Let us denote by $(r^{(k)}, q^{(k)})$ the coordinates of the k-th MAV; the discrete dynamics is given by

$$r_{i+1}^{(k)} = 2r_i^{(k)} - r_{i-1}^{(k)} + \Delta t \left(q_i^{(k)} \cdot \int_i^{i+\Delta t} A^{(k)} dt \cdot (q_i^{(k)})^* + a_g \Delta t \right) \tag{27}$$

$$q_{i+1}^{(k)} = q_i^{(k)} + \frac{1}{2} q_i^{(k)} \cdot \int_i^{i+\Delta t} \Omega^{(k)} dt. \tag{28}$$

Each MAV, in addition to the measurement model (19), may perform relative observation; the general multi robot observation equation can be written as

$$z_i^{(k)} = h^{(k)}(r_i^{(1)}, q_i^{(1)}, ..., r_i^{(k)}, q_i^{(k)}, ..., r_i^{(N)}, q_i^{(N)}). \tag{29}$$

Simple and common examples of relative observations are distance measures. If the k-th MAV measures its own distance from the j-th MAV, the observation is given by

$$z_i^{(k)} = (r_{i,x}^{(k)} - r_{i,x}^{(j)})^2 + (r_{i,y}^{(k)} - r_{i,y}^{(j)})^2 + (r_{i,z}^{(k)} - r_{i,z}^{(j)})^2.$$

4.2.2 The distributed EIF

When the exploration starts, each MAV begins to integrate the information provided by its own sensors by equation (1)-(2) as described before. In particular for any measurement, the incoming data are stored in the bottom-right block of the information matrix and, as a consequence, in the last entries of the information vector:

$$\Sigma_{i-1} \to \Sigma_i = \begin{pmatrix} \Sigma_{i-1} & 0^{7(i-1)\times 7} \\ 0^{7\times 7(i-1)} & 0^{7\times 7} \end{pmatrix} + \begin{pmatrix} 0^{7(i-3)\times 7(i-3)} & 0^{7(i-3)\times 21} \\ 0^{21\times 7(i-3)} & \Sigma_{obs} \end{pmatrix}$$

$$\xi_{i-1} \to \xi_i = \begin{pmatrix} \xi_{i-1} \\ 0^{7\times 1} \end{pmatrix} + \begin{pmatrix} 0^{7(i-3)\times 1} \\ \xi_{obs} \end{pmatrix}.$$

Suppose that after i_1 updating time-steps for the j_1-th MAV and i_2 steps for the j_2-th MAV a relative measurement occurs and for sake of semplicity suppose that $j_1 < j_2$. Each MAV has to increase the size of the information matrix and information vector in order to store the new data. The process is carried out following the steps described below:

1. *State augmentation*. The states of the two MAVs are increased in order to have the same size $7(i_1 + i_2)$; this can be done adding a suitable number of zeros in the information matrix and information vector.

$$\Sigma_{(j_1),i_1} \to \begin{pmatrix} \Sigma_{(j_1),i_1} & 0^{7i_1\times 7i_2} \\ 0^{7i_2\times 7i_1} & 0^{7i_2\times 7i_2} \end{pmatrix}, \qquad \xi_{(j_1),i_1} \to \begin{pmatrix} \xi_{(j_1),i_1} \\ 0^{7i_2\times 1} \end{pmatrix}$$

$$\Sigma_{(j_2),i_2} \to \begin{pmatrix} 0^{7i_1\times 7i_1} & 0^{7i_1\times 7i_2} \\ 0^{7i_2\times 7i_1} & \Sigma_{(j_2),i_2} \end{pmatrix}, \qquad \xi_{(j_2),i_2} \to \begin{pmatrix} 0^{7i_1\times 1} \\ \xi_{(j_2),i_2} \end{pmatrix}$$

2. *Relative estimation*. The information from relative observations are integrated using the standard update equations (1)-(2). Correlation between the estimates on the last poses of the MAVs may appear, so that the updated matrices may be not block-diagonal.

$$\Sigma_{(j_1),i_1} \to \begin{pmatrix} \Sigma_{(j_1),i_1} & * \\ ** & * \end{pmatrix}, \quad \xi_{(j_1),i_1} \to \begin{pmatrix} \xi_{(j_1),i_1} \\ ** \end{pmatrix}$$

$$\Sigma_{(j_2),i_2} \to \begin{pmatrix} * & * \\ ** & \Sigma_{(j_2),i_2} \end{pmatrix}, \quad \xi_{(j_2),i_2} \to \begin{pmatrix} * \\ \xi_{(j_2),i_2} \end{pmatrix}$$

3. *Data fusion*. A communication is established between the MAVs and they exchange their stored data. The data fusion scheme is a non negligible theoretical issue: as a matter of fact, if the process is carried out taking simply the sum of the contributions from each MAV, estimation errors may arise due to adding several times the same information. Following (1), we have adopted a fusion algorithm based on a convex combination of the data:

$$\Sigma_{(j_1),i_1} \rightarrow \omega\Sigma_{(j_1),i_1} + (1-\omega)\Sigma_{(j_2),i_2}, \qquad \xi_{(j_1),i_1} \rightarrow \omega\xi_{(j_1),i_1} + (1-\omega)\xi_{(j_2),i_2}$$

$$\Sigma_{(j_2),i_2} \rightarrow (1-\omega)\Sigma_{(j_1),i_1} + \omega\Sigma_{(j_2),i_2}, \qquad \xi_{(j_2),i_2} \rightarrow (1-\omega)\xi_{(j_1),i_1} + \omega\xi_{(j_2),i_2}$$

As proved in (12), for any $0 < \omega < 1$, the above convex combinations lead to unbiased and consistent estimates, i.e. no overconfident estimate is performed and there is no overlapping of information.

5. Performance Evaluation

In order to validate our approach we have performed simulations. We have considered a conventional scenario defined by a few parameters which regard the robot(s) perception and the environment properties. All our simulations are implemented in Matlab and tested on a computer with 1 Intel Pentium CPU M 1.70GHz, 512MB of memory.

5.1 Single robot SLAM

5.1.1 Simulated scenario

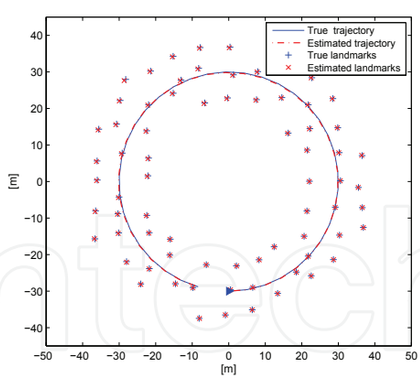
In our simulations we consider a two-wheels robot moving in a $110m$x$110m$ rectangular area in which many point landmarks are randomly distributed. Let us indicate the average landmark density with ρ_L, the robot average speed with v, and the distance traveled by the robot with d. The data associations ere supposed to be given. We consider a robot equipped with wheel encoders which provide the proprioceptive measurements. We base on the Chong-Kleeman ((3)) 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:

$$\begin{aligned} \delta\rho^{R/L} &= \delta\rho^{aR/L}\delta_{R/L} + \nu^{R/L} \\ \nu^{R/L} &\sim N(0, K|\delta\rho^{aR/L}|) \end{aligned} \tag{30}$$

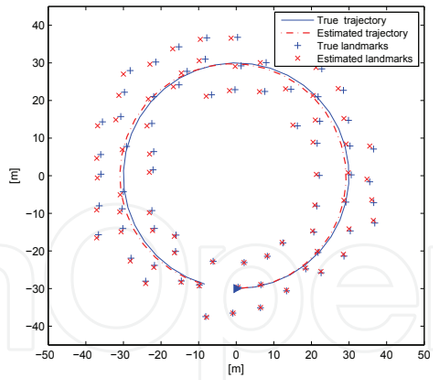
In other words, both $\delta\rho^R$ and $\delta\rho^L$ are assumed Gaussian random variables, whose mean values are given by the actual values (respectively, $\delta\rho^{aR}$ and $\delta\rho^{aL}$), and whose variance also increases linearly with the travelled distance. In our simulation we set $K = 0.00001m$, which corresponds to an indoor environment (19). 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 σ_R^2 and σ_B^2, respectively. The frequency is $0.2Hz$.

5.1.2 Results

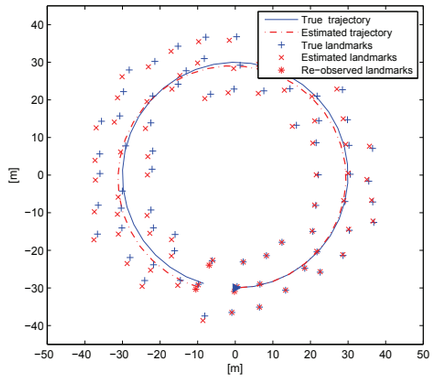
Figures 2(a)-2(b) illustrate the results provided by a given simulation in which the robot closes a loop in counter clockwise direction. Let us point out that the loop closure does not consist of the trajectory closure but it consists of the re-observation of landmarks located close to the starting point. We implemented both our method and the ESDF one. As said in section 3.3, 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 $T_s = 180s$.
Figures 2(a) and 2(b) 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



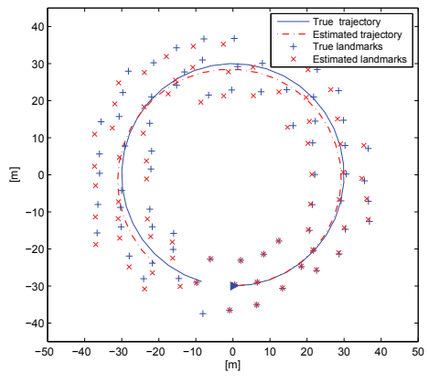
(a) Simulation 1: estimation results of our method before the loop closure.



(b) Simulation 1: ESDF estimation results before the loop closure.



(c) Simulation 1: estimation results of our method after the loop closure.



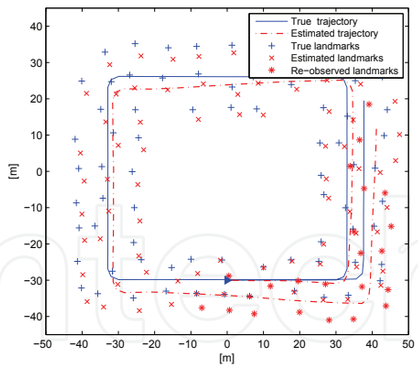
(d) Simulation 1: ESDF estimation results after the loop closure.

Fig. 2. Pictures for the results of Simulation 1

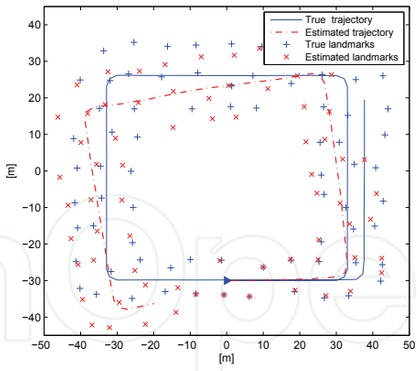
solid blue line and cross blue markers, respectively. Moreover, the estimated trajectory and landmark locations are represented by a dash-dotted red line and red x-markers, respectively. In order to provide quantitative results, we consider the error on the estimated map by computing for all the landmarks 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.

Figures 2(c) and 2(d) 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.

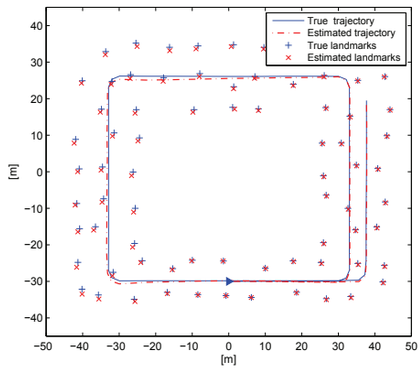
The results provided by a second simulation are shown in Figures 3(a)-3(b): the robot closes a loop in counter clockwise direction and than goes on re-traversing a region for a long time. The parameters of this simulation are: $\sigma_B = 1deg$, $\sigma_R = 1cm$, $\rho_L = 0.02m^{-2}$, $v = 1.2ms^{-1}$,



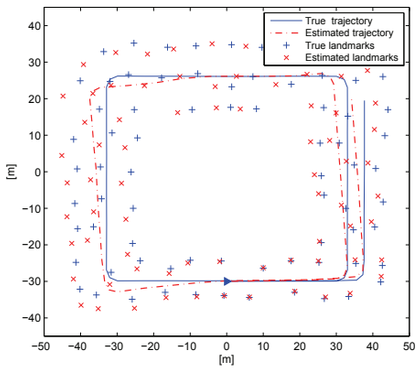
(a) Simulation 2: estimation results of our method before the loop closure.



(b) Simulation 2: ESDF estimation results before the loop closure.



(c) Simulation 2: estimation results of our method after the loop closure.



(d) Simulation 2: ESDF estimation results after the loop closure.

Fig. 3. Pictures for the results of Simulation 2

$d = 325m$. The simulation time is $T_s = 270s$. Figs. 3(a) and 3(b) show the two methods before exploiting the loop closure information. Concerning our method, as said in section 3.3, 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 3-(a) 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 3(c) 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 3(d) 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.

5.2 Cooperative MAV localization

5.2.1 The simulated environment

The trajectories of the MAVs are generated randomly and independently one each other. In particular, for every MAV, the motion is generated by generating randomly the linear and angular acceleration at $100Hz$. Specifically, at each time step, the three components of the linear and the angular acceleration are generated as Gaussian independent variables with mean values μ_a and $\mu_{\dot{\Omega}}$ and with covariance matrices P_a and $P_{\dot{\Omega}}$. By performing many simulations we remarked that the precision of the proposed strategy is almost independent of all these parameters. The simulations provided in this section are obtained with the following settings: $\mu_a = \mu_{\dot{\Omega}} = [000]^T$,

$$P_a = \begin{bmatrix} (5ms^{-2})^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and

$$P_{\dot{\Omega}} = \begin{bmatrix} (10deg\ s^{-2})^2 & 0 & 0 \\ 0 & (10deg\ s^{-2})^2 & 0 \\ 0 & 0 & (10deg\ s^{-2})^2 \end{bmatrix}$$

We adopt many different values for the initial MAV positions orientations and speeds. We also consider different scenarios corresponding to a different number of MAVs.
Starting from the accomplished trajectories, the true angular speed and the linear acceleration are computed at each time step of $0.01s$ (respectively, at the time step i, we denote them with Ω_i^{true} and A_i^{true}). Starting from them, the IMU sensors are simulated by generating randomly the angular speed and the linear acceleration at each step according to the following: $\Omega_i = N\left(\Omega_i^{true}, P_{\Omega_i}\right)$ and $A_i = N\left(A_i^{true} - A_{g\,i}, P_{A_i}\right)$ where N indicates the Normal distribution whose first entry is the mean value and the second one its covariance matrix and P_{Ω_i} and P_{A_i} are the covariance matrices characterizing the accuracy of the IMU; finally, A_g is the gravity acceleration expressed in the local frame. In all the simulations we set both P_{A_i} and P_{Ω_i} diagonal matrices. In the results here provided they are set as follows:

$$P_{A_i} = \begin{bmatrix} (0.1ms^{-2})^2 & 0 & 0 \\ 0 & (0.1ms^{-2})^2 & 0 \\ 0 & 0 & (0.1ms^{-2})^2 \end{bmatrix}$$

and

$$P_{\Omega_i} = \begin{bmatrix} (10deg\ s^{-1})^2 & 0 & 0 \\ 0 & (10deg\ s^{-1})^2 & 0 \\ 0 & 0 & (10deg\ s^{-1})^2 \end{bmatrix}$$

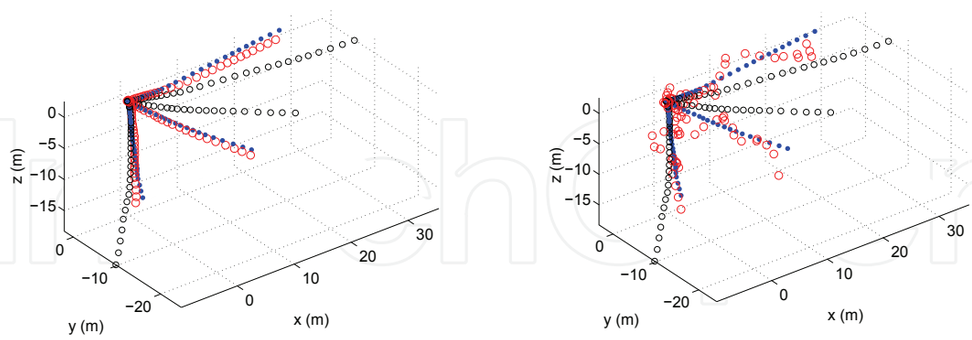
for every step i.
The MAVs are also equipped with GPS and range sensors. The GPS provides the position of the MAV with a Gaussian error whose covariance is a diagonal matrix and whose components are equal to $25m^2$. The GPS data are delivered at $5Hz$. Finally, the range sensors provide the distances among the MAVs at $2Hz$ and the measurement errors are normally distributed with variance $(0.01m)^2$.

5.2.2 Results

We provide some of the results obtained with the previous settings and by simulating N MAVs. In particular, we consider the case of $N = 3$ and $N = 5$. Furthermore, we consider separately the cases when the estimation is performed by only integrating the IMU data, by combining the IMU data with the GPS data and by combining all the sensor data. Finally, in order to evaluate the benefit of using the projection filter discussed in Section 4.1.2, we consider separately the cases when this filter is adopted and when it is not adopted.

Figs. 5(a)-5(b) show the results obtained with three MAVs. The blue dots represent the ground truth. In Fig. 5(a) the magenta dots represent the GPS data and the black circles the trajectories estimated by only integrating the IMU data. It is clear that both IMU and GPS are very noisy and cannot be used separately to estimate the MAV trajectories. In Fig. 5(b) the green dots represent the trajectories estimated by fusing the IMU data and the GPS data with our proposed approach (EIF and projection filter). Finally, the red dots represent the result obtained by also fusing the range measurements. We remarked that the use of the range measurements further reduce the error. In particular, for the simulation in fig. 1a-b the position error averaged on all the three MAV and on all the time steps is equal to $0.6m$ without the range measurements and $0.45m$ with them. As expected, this improvement is still larger by increasing the number of MAVs. In Figs. 4(c)-4(d) the results obtained by using 5 MAVs is shown. The position error obtained by also fusing the range measurements reduces to $0.2m$.

Fig. 4 shows the benefit of using the Projection filter discussed in Section 4.1.2. In particular, in Fig. 4(a) the red circles represent the trajectories estimated by fusing all the sensor data and by running the Projection Filter at $5Hz$ while in Fig. 4(b) the red circles represent the trajectories estimated without the use of the Projection Filter. As in the previous figures, the ground truth is represented with blue dots and the black dots represent the trajectories obtained by a simple integration of the IMU data.



(a) (b)

Fig. 4. Blue points represent true MAVs trajectories, balck circles are the estimated trajectories via odometry and red circles are the estimated trajectories with the EIF. Figure 4(a) represents the simulation of a 3-MAV system; Figure 4(b) represents the same scenario without taking into account the information provided by the projection filter.

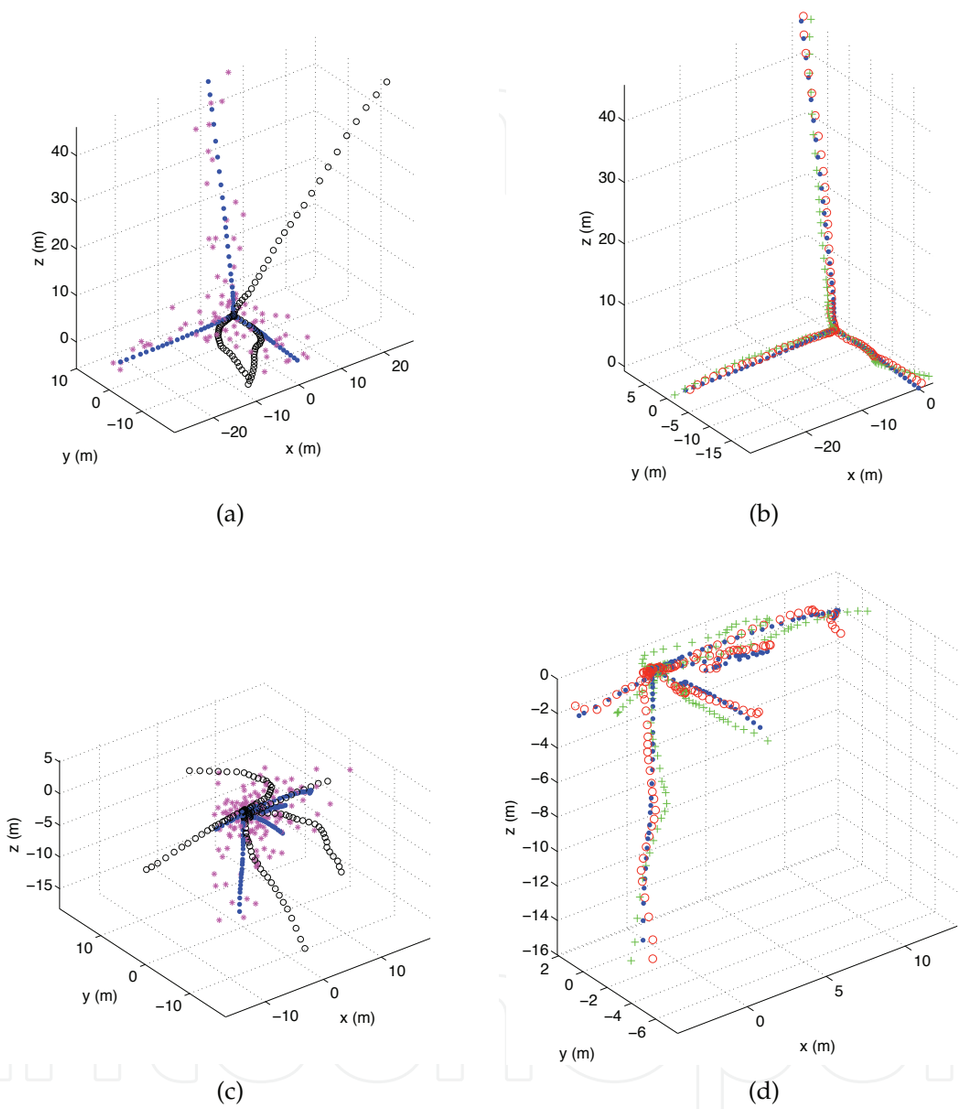


Fig. 5. Blue points represent true MAVs trajectories, black circles are the trajectories with only odometric estimates, magenta stars are the GPS data, green stars are the trajectory estimates without taking into account relative observations and red circles are the estimates with the complete distributed EIF. Figures 5(a)-5(b) are the simulation of 3-MAV scenario, while in Figures 5(c)-5(d) is plotted the evolution of a 5-MAV system.

6. Conclusions

In this chapter we considered the problem of cooperative localization and SLAM by using an Extended Information Filter.
We started by considering the ESDF technique (6) 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.
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.
Furthermore, we considered the cooperative case, i.e. when the estimation process is performed simultaneously by a team of agents. In this case, two original contributions have been introduced. The former consists of a simple trick which allowed us to avoid the equations which characterize the prediction phase of the extended information filter. In particular, the information contained in the data provided by the inertial sensors is exploited by using the equations which characterize the perception step of the EIF. This allowed us to easily distributing the entire estimation process over all the team members. The latter contribution is the use of a projection filter which allowed exploiting the information contained in the geometrical constraints which arise as soon as the MAV orientations are characterized by unitary quaternions.
The performance of the proposed approaches has been evaluated by using synthetic data.

7. Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 231855 (sFly).

8. References

[1] J. Capitan, L. Merino, F. Caballero and A. Ollero, Delayed-state Information Filter for Cooperative Decentralized Tracking, *ICRA*, 2009, Kobe, Japan, pages 3865-3870
[2] J.A. Castellanos, J. Neira and J.D. Tardos, Limits to Consistency of the EKF-based SLAM, *Intelligent Autonomous Vehicle*, 2004, Lisbon, Portugal
[3] K.S. Chong and L. Kleeman, Accurate Odometry and Error Modelling for a Mobile Robot, *IEEE International Conference on Robotics and Automation (ICRA97)*, 1997, Albuquerque, New Mexico, USA
[4] Crowley, J.L., (1989). World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. IEEE International Conference on Robotics and Automation (ICRA), Scottsdale, AZ
[5] Dissanayake, Newman, Clark, Durrant-Whyte and Csorba, 2001, A Solution to the Simultaneous Localization and Map Building (SLAM) problem, IEEE Trans. On Rob. And Aut. Vol 17, No.3, June 2001
[6] R.M. Eustice, H. Singh and J.J. Leonard, Exactly Sparse Delayed-State Filters for View-Based SLAM, *IEEE Trans. on Robotics*, vol. 22, n. 6, 2006, pages 1100-1114
[7] D. Fox, W. Burgard, H. Kruppa, S. Thrun, 2000, A Probabilistic Approach to Collaborative Multi-Robot Localization, Autonomous Robots 8, 2000, pages 325-344
[8] R. Grabowski, L.E. Navarro-Serment, C.J.J. Paredis, P.K. Khosla, 2000, Heterogeneous

Teams of Modular Robots for Mapping and Exploration, Autonomous Robots, Vol. 8, n. 3, June 200?, pages 325-344

[9] G. Grisetti, C. Stachniss, s. Grzonka and W. Burgard, A Tree Parametrization for Efficiently Computing Maximum Likelihood Maps Using Gradient Descent, *Robotics: Science and Systems (RSS07)*, 2007, Atlanta, Georgia, USA

[10] A. Howard, M.J. Mataric and G.S. Sukhatme, "Localization for Mobile Robot Teams Using Maximum Likelihood Estimation", *International Conference on Inteligent Robot and Systems (IROS02)*, Volume: 3 , 30 Sept.-5 Oct. 2002,Lausanne, pages 2849-2854

[11] E. Jones, A. Vedaldi and S. Soatto, Inertial Structure from Motion with Autocalibration, *ICCU Workshop*, 2007

[12] S.J. Julier and J.K. Uhlmann, A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations, *American Control Conference*, 1997, Albuquerque, New Mexico, pages 2369-2373

[13] S.J. Julier and J.K. Uhlmann, A Counterexample to the Theory of Simoultaneous Localization and Map Building, *IEEE International Conference on Robotics and Automation (ICRA01)*, 2001, Seoul, Korea

[14] M. Kaess, A. Ranganathan and F. Dellaert, iSAM: Incremental Smoothing and Mapping, *IEEE Trans. on Robotics*, vol. 24, n.6, 2008, pages 1365-1378

[15] K. Kato, H. Ishiguro, M. Barth, "Identifying and Localizing Robots in a Multi-Robot System Environment" *International Conference on Inteligent Robot and Systems (IROS99)* 1999

[16] J.J. Leonard, H.F. Durrant-Whyte, "Directed Sonar Sensing for Mobile Robot Navigation," *Kluwer Academic Publishers*, Dordrecht, 1992

[17] A. Martinelli, F. Pont and R. Siegwart , "Multi-Robot Localization Using Relative Observations" *IEEE International Conference on Robotics and Automation (ICRA05)*, 2005, Barcellona, Spain

[18] A. Martinelli and R. Siegwart, Exploiting the Information at the Loop Closure in SLAM, *IEEE International Conference on Robotics and Automation (ICRA07)*, 2007, Rome, Italy

[19] A. Martinelli, N. Tomatis and R. Siegwart, Simultaneous Localization and Odometry Self-Calibbration for Mobile Robot, *Autonomous Robots*, 2007, 22, pp. 75-85

[20] A.I. Mourikis, S.I. Roumeliotis, "Optimal Sensing Strategies for Mobile Robot Formations: Resource-Constrained Localization", Robotics: Science and Systems June 8-11, 2005 Massachusetts Institute of Technology Cambridge, Massachusetts, USA

[21] P. Newman, On the structures and solution of simultaneous localization and mapping problem, PhD thesis, Australian Center for Field Robotics, Sidney, 1999

[22] E. Olson, J. Leonard and S. Teller, Fast Iterative Alignment of Pose Graphs with Poor Initial Estimates, *IEEE International Conference on Robotics and Automation (ICRA06)*, 2006, Orlando, Florida, USA

[23] I.M. Rekleitis, G. Dudek and E.E. Milios, "Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy " *International Conference on Inteligent Robot and Systems (IROS02)* Lausanne, Switzerland

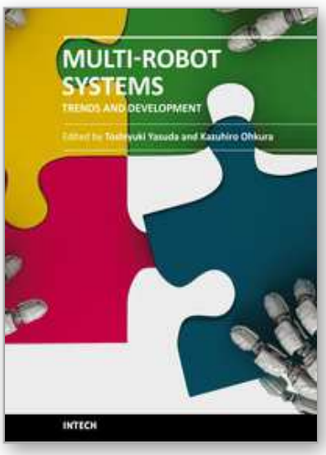
[24] S.I. Roumeliotis and G.A. Bekey, 2002, Distributed Multirobot Localization, IEEE Transaction On Robotics And Automation Vol 18, No.5, October 2002

[25] J. Schewchuk, An Introduction to Conjugate Gradient Method without Agonizing Pain, Carnegie-Mellon University, Pittsburgh, PA, Tech. Report CMU-CS-94-125, 1994

[26] Smith, Self, et al. (1988) "Estimating uncertain spatial relationships in robotics" *Uncertainty in Artificial Intelligence 2* Elsevier Science Pub: 435-461

[27] J.R. Spletzer and C.J. Taylor, "A Bounded Uncertainty Approach to Multi-Robot Localization" *International Conference on Inteligent Robot and Systems (IROS03)* Las Vegas, USA, 2003

[28] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, MIT Press, Cambridge, MA, 2005



Multi-Robot Systems, Trends and Development

Edited by Dr Toshiyuki Yasuda

ISBN 978-953-307-425-2
Hard cover, 586 pages
Publisher InTech
Published online 30, January, 2011
Published in print edition January, 2011

This book is a collection of 29 excellent works and comprised of three sections: task oriented approach, bio inspired approach, and modeling/design. In the first section, applications on formation, localization/mapping, and planning are introduced. The second section is on behavior-based approach by means of artificial intelligence techniques. The last section includes research articles on development of architectures and control systems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Francesco Conte, Andrea Cristofaro, Alessandro Renzaglia and Agostino Martinelli (2011). Cooperative Localization and SLAM Based on the Extended Information Filter, Multi-Robot Systems, Trends and Development, Dr Toshiyuki Yasuda (Ed.), ISBN: 978-953-307-425-2, InTech, Available from: http://www.intechopen.com/books/multi-robot-systems-trends-and-development/cooperative-localization-and-slam-based-on-the-extended-information-filter

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen