Distributed Localization and Submapping for Robot Formations using a prior map ⋆

María T. Lázaro, Lina M. Paz, Pedro Piniés, José A. Castellanos

Instituto de Investigación en Ingeniería de Aragón,
Universidad de Zaragoza, Spain
E-mail:[mtlazaro, linapaz, ppinies, jacaste]@ unizar.es

Abstract: In this paper, we present a distributed algorithm which allows a robot formation to jointly improve a prior stochastic map of the environment where it has to accomplish a commanded task. To reduce the computational cost of the global map updates we exploit the fact that a robot formation work in the same map area allowing us to use conditional independence properties over the state distribution. Each robot maintains its own local and global maps which can be improved with the information received when communications among robots take place. Besides, the robots also exchange their positions in order to maintain the structure of the formation. Simulation experiments were conducted showing that, after the synchronization steps, each robot will have exactly the same information about the map and about the location of the robots at its disposal. Our results also demonstrate the achieved precision and efficiency of the proposed distributed algorithm.

Keywords: Robot formations, distributed SLAM.

1. INTRODUCTION

The fact that a team of robots could develop its tasks autonomously could be of crucial importance in hostile environments of hard access to humans. In some missions such as rescue operations, these robots need to maintain a formation structure, i.e. a geometrical configuration. One of the main potential applications of formations in emergency situations is to guide a group of people to a safe exit while protecting them on their way out (Urcola and Montano (2011)). Therefore, each robot must know the position of the other members of the team, and also understand the surrounding environment to avoid dangerous situations. Decentralized calculations improve the robustness of the system in case of communication loss or member failures. When a prior map of the rescue area is available the success of the mission increases since the robots can plan routes and take decissions in advance. In this case robots have to update the prior map to cope with changes encountered during the mission.

According to the previous description, the main goal of this paper is a novel and efficient algorithm to localize a robot formation and improve a given prior stochastic map using a distributed paradigm. Each robot updates its own copy of the prior global map by sending/receiving local information to/from the rest of the team. Our distributed estimation algorithm is described in terms of a Gaussian Markov Random Filed (GMRF) which allows us to analyze the conditional independence (CI) properties of the problem to achieve high efficiency (Bishop (2006)). The application of the CI property in the algorithm is twofold. On the one hand, the robots in the formation only need to constantly update the local region in which they move whereas global updates can be postponed reducing computational cost (Piniés and Tardós (2008)). On the other hand, robots do not make observations of each other in our system but get indirectly related by observing common map features. The load in the communication channel is then reduced since each robot just sends an information summary of features observed since last communication. As a result, the algorithm proposed does not rely on a central server improving flexibility and robustness and reduces the computational and communication requirements at each step. Moreover, it is shown that, in a linear filtering context, the resulting decoupled method produces the exact results than using only one filter. In this paper, we use the Extended Information Filter (EIF) as the core of the distributed algorithm such that after all messages are sent/received each robot estimate is equal to the centralized solution.

The problem of localizing a robot formation within a prior map was previously treated in Lázaro and Castellanos (2010). In that work, a centralized localization process at the leader robot is proposed but the prior map remains unmodified. In Bailey et al. (2011), a distributed EIF algorithm to jointly localize a team of robots is presented. In Roumeliotis and Bekey (2002) an Extended Kalman Filter (EKF) is used whereas in Nerurkar et al. (2009) the authors implement a Maximum a Posteriori (MAP) estimator. Unlike these previous works, we do not use inter-robot measurements to improve the location of the team but a prior map that is mantained and improved.

In Grime and Durrant-Whyte (1994) the Channel Filter (CF) is introduced for the solution of general distributed problems. The CF prevents double-counting of information by using a tree communication topology and by keeping a record of the information transmitted over the communica-

⋆ This work was supported by the Spanish projects MICINN-FEDER DPI2009-13710 and DPI2012-36070.

tion channel. The transmission of information in our algorithm is similar to this filter but we avoid double-counting by synchronizing the transmission of messages. The main difference with that work is that we implement in addition an efficient algorithm to update the state vectors of the robots by postponing global updates without introducing approximations. In the same multi-robot context, recent works also address the distributed Simultaneous Localization and Mapping (SLAM) problem. In Leung et al. (2011) each robot computes its own centralized-equivalent estimate of the system using the odometry and measurements that have been received from other robots. In Cunningham et al. (2010), the authors formulate the problem as a non linear least squares optimization where robots asynchronously send condensed landmarks information. The work presented in Nettleton et al. (2003), copes with bandwidth communication requirements, in such a way that each robot selects a set of features with the greatest information gain to be sent. To avoid the double counting information problem, they combine an EIF with the Convariance Intersection algorithm, a sub-optimal filter which guarantees consistency although yielding pessimistic estimates. In contrast to these works, in our approach the robots only communicate their positions and the new local information gathered since last synchronization using small local matrices that reduce the communication bandwidth.

The outline of the paper is as follows: Section 2 describes the main steps and notation of our distributed localization and mapping algorithm for robot formations with a prior map. Section 3 brings a summarized description of the conditional independence property of a Markov Random Field. Section 4 makes use of the CI property to reduce the computational cost for each robot when working on a local region of the global map. Section 5 is devoted to explain the message passing protocol which also takes into account the CI property. The complete distributed localization and mapping algorithm is described in section 6 together with a computational and communication complexity analysis. In section 7 the results of the testing experiments are presented and finally in section 8, we draw some conclusions and future work is proposed.

2. PROBLEM STATEMENT

Given a previously built feature-based stochastic map $\mathbf{x}_F$, our goal is to localize a formation of n_r robots in it while improving the map estimate. The robot formation and the map are given by $[\mathbf{x}_\mathcal{R}^T \mathbf{x}_F^T]^T$, where $\mathbf{x}_\mathcal{R} = \{\mathbf{x}_R^j | j = 1 \dots n_r\}$ contains the location of the team of robots.

A naive approach to update the joint estimate is to use a centralized method where a leader fuses all the odometry and sensor measurements gathered by the rest of the team. Instead, we implement a distributed algorithm in which each robot j estimates its own pose $\mathbf{x}_R^j$ and updates its own estimate of the map $\mathbf{x}_F^j$. Robots periodically broadcast their position to maintain the shape of the formation while navigating towards the goal. During the periods of lack of communication, each robot predicts the position of the rest of the team based on its knowledge about the formation structure. The advantage of this distributed strategy over the centralized one is that the system becomes more robust since each robot keeps an estimate of its position and the global map and does not depend on the availability of a leader or constant communication with the rest of the team to maintain the shape of the formation.

The efficiency of the algorithm proposed in terms of computational cost and communication bandwidth is based on the following two ideas:

(1) In order to reduce the computational cost, the prior map $\mathbf{x}_F$ is divided into local working areas. While the formation traverses a local region $\mathbf{x}_{F_l}$ each robot updates the features in it using the measurements gathered at each step whereas the rest of the map $\mathbf{x}_{F_g}$ is not modified. When the formation moves to a new local area all robots update the previous unmodified features $\mathbf{x}_{F_g}$, assign the new set of local landmarks to $\mathbf{x}_{F_l}$ and repeats the same procedure. Since the number of features in a local region is bounded, this algorithm maintains a constant computational cost $\mathcal{O}(1)$ when working in a local area. In addition, we will show that the map estimate obtained is the same as if we had been working with the whole map.

(2) When working in a local region, robots send messages to each other at synchronization steps to improve their own local estimates by using the information gathered by other members of the team. During these synchronization steps, we assume all-to-all communication availability. Instead of transmitting raw measurements, robots send information matrices of the features observed since last synchronization reducing the amount of information on the communication channel. After each synchornization step, each robot has the same information such that the map estimates coincide with the one obtained using a centralized version, i.e. $\mathbf{x}_F^j \to \mathbf{x}_F | j = 1 \dots n_r$.

Previous ideas are based on the CI property of the variables involved in the estimation. Thus, in the next section we give a brief and intuitive review of this property.

3. THE CONDITIONAL INDEPENDENCE PROPERTY

We will refer to the example in Fig. 1 where a *Markov Random Field* (MRF) is used to show the CI property of a set of random variables $\mathbf{x}_A$, $\mathbf{x}_B$ and $\mathbf{x}_C$. Suppose that we want to search for a path connecting any node in $\mathbf{x}_A$ to any node in $\mathbf{x}_B$ when the common node subset $\mathbf{x}_C$ is removed from the graph. Since no such a path exists, we can assert that subsets $\mathbf{x}_A$ and $\mathbf{x}_B$ are conditionally independent if we know the subset $\mathbf{x}_C$. Then, the CI property is determined by simple graph separation (Bishop (2006)). Formally this is expressed by any of the following equivalent expressions:

$$p(\mathbf{x}_A, \mathbf{x}_B | \mathbf{x}_C) = p(\mathbf{x}_A | \mathbf{x}_C) p(\mathbf{x}_B | \mathbf{x}_C) \quad (1)$$

$$p(\mathbf{x}_A | \mathbf{x}_B, \mathbf{x}_C) = p(\mathbf{x}_A | \mathbf{x}_C) \quad (2)$$

$$p(\mathbf{x}_B | \mathbf{x}_A, \mathbf{x}_C) = p(\mathbf{x}_B | \mathbf{x}_C) \quad (3)$$

These equations will be applied in the explanation of the two following sections. Along each description we will make use of a MRF similar to Fig. 1 to help the reader recognize the CI-property.

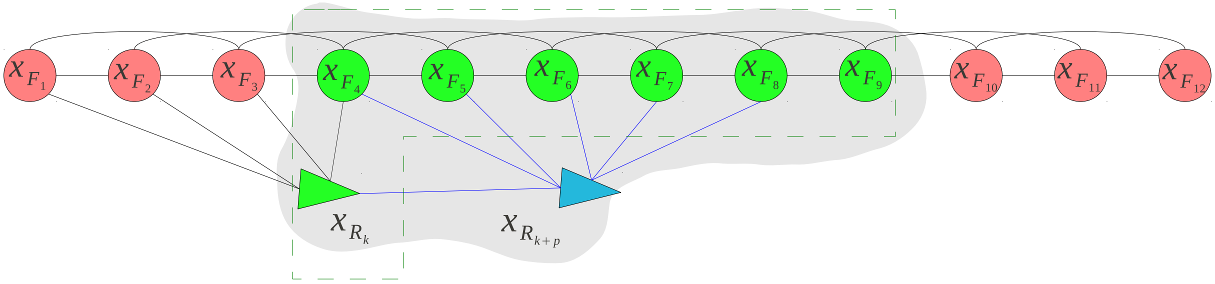

Fig. 2. GMRF of the individual robot estimation process. The local map corresponds to elements inside the shadowed region. Since there is no link between robot $\mathbf{x}_{R_{k+p}}$ and features F_g, the initial robot position at instant k and the local features $F_l = \{F_4 \dots F_9\}$ make the robot at instant $k+p$ conditionally independent of map elements $F_g = \{F_1 \dots F_3, F_{10} \dots F_{12}\}$. To easily verify the CI-property node colors have been selected to match those in Fig.1. Also, the common separator is surrounded by a dash line.

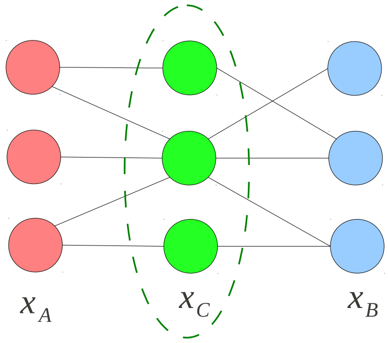

Fig. 1. This example shows a MRF with no direct links from subset $\mathbf{x}_A$ to subset $\mathbf{x}_B$. This means that the conditional independence property $p(\mathbf{x}_A, \mathbf{x}_B|\mathbf{x}_C) = p(\mathbf{x}_A|\mathbf{x}_C)p(\mathbf{x}_B|\mathbf{x}_C)$ or any of its equivalents holds for any probability distribution described by this graph. Notice that we do not make any assumption about the nature of the distribution. Common separator nodes are surrounded by a dash dark line and depicted in green; nodes in $\mathbf{x}_A$ and $\mathbf{x}_B$ sets are shown in red and blue respectively.

4. LOCAL ESTIMATES AND GLOBAL UPDATES

In this section we focus on how each robot works in a local region and is able to update the prior map. At instant k, the state vector for a robot is given by $[\mathbf{x}_{R_k}^T \mathbf{x}_F^T]^T$, i.e the current robot position and the estimate of the map. During this explanation we will omit the robot super index to simplify the notation. We assume a multivariable Gaussian distribution on the state vector described by the information vector $\mathbf{i}_k$ and matrix $\mathbf{I}_k$.

In order to work in local regions with a consistent procedure for global map updates, we distinguish three operations: GlobalToLocal, Local EIF and LocalToGlobal. We will use the Gaussian Markov Random Field in Fig. 2 to show the application of the conditional independence property in these operations.

4.1 GlobalToLocal

The prior map of the example is composed by feature elements $\mathbf{x}_F = \{\mathbf{x}_{F_1} \dots \mathbf{x}_{F_{12}}\}$. Instead of working with the whole map $\mathbf{x}_F$ we want to work in the local region shadowed in the figure. At instant k, the robot $\mathbf{x}_{R_k}$ is about to enter to this local region from an already updated global map whose information matrix and vector are given by $\mathbf{I}_k$ and $\mathbf{i}_k$ respectively. The new local state vector will be $\mathbf{x}_l = [\mathbf{x}_{R_k}^T, \mathbf{x}_{F_l}^T]^T$ formed by the current robot position and the feature subset $F_l = \{F_4 \dots F_9\}$. Features that are not in the local region correspond to elements $F_g = \{F_1 \dots F_3, F_{10} \dots F_{12}\}$. To obtain the local region $\mathbf{x}_l$ we just marginalize it from the joint distribution as shown in Eq. (4),

$$p(\mathbf{x}_{R_k}, \mathbf{x}_{F_l}) = \int p(\mathbf{x}_{R_k}, \mathbf{x}_{F_l}, \mathbf{x}_{F_g}) d\mathbf{x}_{F_g} \tag{4}$$

For Gaussian distributions the marginal is given by the Schur complement (Bishop (2006)), obtaining the local marginal information $\mathbf{i}_{l,k}^m, \mathbf{I}_{l,k}^m$ from the global state $\mathbf{i}_k, \mathbf{I}_k$ as it is shown in Algorithm 1. A copy of the marginal at instant k is stored for future use in the distributed algorithm.

Algorithm 1 $(\mathbf{I}_{l,k}^m, \mathbf{i}_{l,k}^m)$ = **GlobalToLocal** $(\mathbf{I}_k, \mathbf{i}_k)$

$$\mathbf{I}_k = \begin{pmatrix} \mathbf{I}_l & \mathbf{I}_{lg} \\ \mathbf{I}_{gl} & \mathbf{I}_g \end{pmatrix}, \mathbf{i}_k = \begin{pmatrix} \mathbf{i}_l \\ \mathbf{i}_g \end{pmatrix} \tag{5}$$

{*Marginalization of local submap elements from the global state*}

$$\mathbf{I}_{l,k}^m = \mathbf{I}_l - \mathbf{I}_{lg}\mathbf{I}_g^{-1}\mathbf{I}_{gl} \tag{6}$$
$$\mathbf{i}_{l,k}^m = \mathbf{i}_l - \mathbf{I}_{lg}\mathbf{I}_g^{-1}\mathbf{i}_g$$

{*Return Marginal distribution at instant k*}

4.2 Local EIF

To operate in the local region $\mathbf{x}_l$ the robot just carries out a standard EIF algorithm (Walter et al. (2007)) in which the initial robot pose is kept in the state vector (i.e. $\mathbf{x}_{R_k}$ is not marginalized out). After p steps the distribution of the global map can be factorized as follows:

$$\begin{aligned} & p(\mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_F|\mathbf{z}_{1:k+p}) = \\ & \quad p(\mathbf{x}_{F_g}|\mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_{F_l}, \mathbf{z}_{1:k+p}) \\ & \quad p(\mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_{F_l}|\mathbf{z}_{1:k+p}) \end{aligned} \tag{7}$$

The second factor $p(\mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_{F_p}|\mathbf{z}_{1:k+p})$ corresponds to the probability distribution of the local region. Notice

Algorithm 2 $(\mathbf{I}_{k+p}, \mathbf{i}_{k+p})$ = **LocalToGlobal** $(\mathbf{I}^m_{l,k}, \mathbf{i}^m_{l,k}, \mathbf{I}^m_{l,k+p}, \mathbf{i}^m_{l,k+p}, \mathbf{I}_k, \mathbf{i}_k)$

{*Information due to the measurements obtained during last p steps*}

$$\mathbf{i}^{new}_{l,k+1:k+p} = \mathbf{i}^m_{l,k+p} - \mathbf{i}^m_{l,k} \quad (9)$$
$$\mathbf{I}^{new}_{l,k+1:k+p} = \mathbf{I}^m_{l,k+p} - \mathbf{I}^m_{l,k}$$

{*Global Update: adding new information*}

$$\mathbf{I}_{k+p} = \mathbf{I}_k \boxplus \mathbf{I}^{new}_{l,k+1:k+p} \quad (10)$$
$$\mathbf{i}_{k+p} = \mathbf{i}_k \boxplus \mathbf{i}^{new}_{l,k+1:k+p} \quad (11)$$

{*Marginalize $\mathbf{x}_{R_k}$ out, the oldest position*}

$(\mathbf{I}_{k+p}, \mathbf{i}_{k+p}) = \texttt{marginalizeOut}(\mathbf{x}_{R_k}, \mathbf{I}_{k+p}, \mathbf{i}_{k+p})$

{*Return Global state vector at instant $k+p$*}

that there is no direct link between $\mathbf{x}_{F_g}$ and $\mathbf{x}_{R_{k+p}}$ in Fig. 2 therefore the first factor can be simplified by using the CI-property as follows:

$$p(\mathbf{x}_{F_g}|\mathbf{x}_{R_{k+p}}, \mathbf{x}_{R_k}, \mathbf{x}_{F_l}, \mathbf{z}_{1:k+p}) = p(\mathbf{x}_{F_g}|\mathbf{x}_{R_k}, \mathbf{x}_{F_l}, \mathbf{z}_{1:k}) \quad (8)$$

Consequently, when the robot performs the move-sense local cycle, the global part $\mathbf{x}_{F_g}$ remains conditionally independent of the current local robot $\mathbf{x}_{R_{k+p}}$ and the new observations $\mathbf{z}_{k+1:k+p}$ that have been gathered in the local region and therefore does not require continous updates.

4.3 LocalToGlobal

When the robot is about to change to a new local region at step $k+p$, it first updates the elements of its total map $\mathbf{i}_k$ and $\mathbf{I}_k$. We take advantage again of the fact that the new local information acquired during last p steps only affects the elements that correspond to the local region. Therefore, features in $\mathbf{x}_{F_g}$ are conditionally independent of measurements $\mathbf{z}_{k+1:k+p}$ (Eq. (8)). In terms of EIF, this statement allows us to easily recover the new information given by $\mathbf{I}^{new}_{l,k+1:k+p}, \mathbf{i}^{new}_{l,k+1:k+p}$ (Algorithm 2, Eq. (9)) from the subtraction of the current local map at instant $k+p$ and the local map at instant k augmented with zeros at the position of robot $\mathbf{x}_{R_k+p}$ to fit the dimensions. This information is finally added to $\mathbf{i}_k$ and $\mathbf{I}_k$, the global state at step k, to update the total estimate. Eq. (10) shows this operation where $\boxplus$ is the operator in charge of adjusting the dimensions of the matrices and vectors for a coherent addition of information to the common local elements. Once the total estimate is updated, we marginalize out the oldest robot position $\mathbf{x}_{R_k}$. Similarly to Eq. (6), we use the Schur Complement to perform the marginalization.

5. PASSING MESSAGES BETWEEN THE ROBOT FORMATION

We will make use of the CI-property to efficiently send and receive update messages between the team of robots. The key idea is that as the robots do not observe each other they just get related by measurements of common map features. This indirect relation means that the robot formation is CI given the map. Formally this insight is represented by the following equation:

$$p(\mathbf{x}^1_{R_k}, \ldots \mathbf{x}^{n_r}_{R_k}|\mathbf{x}_F) = p(\mathbf{x}^1_{R_k}|\mathbf{x}_F) \ldots p(\mathbf{x}^{n_r}_{R_k}|\mathbf{x}_F) \quad (12)$$

In Fig. 3 we can see a small example that shows this property. In the example (figure left) three robots use information of their individual odometry to move from instant k to $k+1$. They also get connected to some map features given the observations at both steps. Since we are using a filtering paradigm, robot positions at k are marginalized out creating a new clique (Eustice et al. (2006)) with all the elements that were connected to them. The result is a new graph (figure right) that links the current positions with map features. Observe that there are no direct links between robots, that is, the robot formation is CI given the map. In subsequent steps this property remains.

In order to obtain the same estimation as in a centralized system, robots are synchronized periodically and broadcast the new information gathered since the last synchronization. From the point of view of a robot, the synchronization is based on two steps: first, the robot broadcasts its own information to the rest of the team; second, it receives messages from the other members of the formation. We explain these steps in the following subsections.

5.1 SendMessages

Algorithm 3 details the operations performed to send a message from robot j to the formation. Suppose that at time s a synchronization occurred and the estimate kept by each robot is updated and coincides with the one obtained in a centralized system. From this recently updated estimate, robot j calculates the marginal of the features $\mathbf{I}^{m,j}_{F,s}$ and $\mathbf{i}^{m,j}_{F,s}$.

When a new synchonization step takes place at $s+p$ the new map information gathered by the robot since last synchronization is calculated. This new information will be the difference between the feature marginals at $s+p$ and s. Eq. (13) in Algorithm 3 shows this operation. The subtracted information is finally broadcasted from robot j to the rest of the team.

5.2 ReceiveMessages

Algorithm 4 details the operations carried out when messages are received to update the map of robot j. As robots are CI given the map, the information sent to robot j only affects its feature elements in the information matrix and vector and therefore can be directly added by using the operator $\boxplus$, as it was explained in subsection 4.3. After these operations, all robots share the same information about the map.

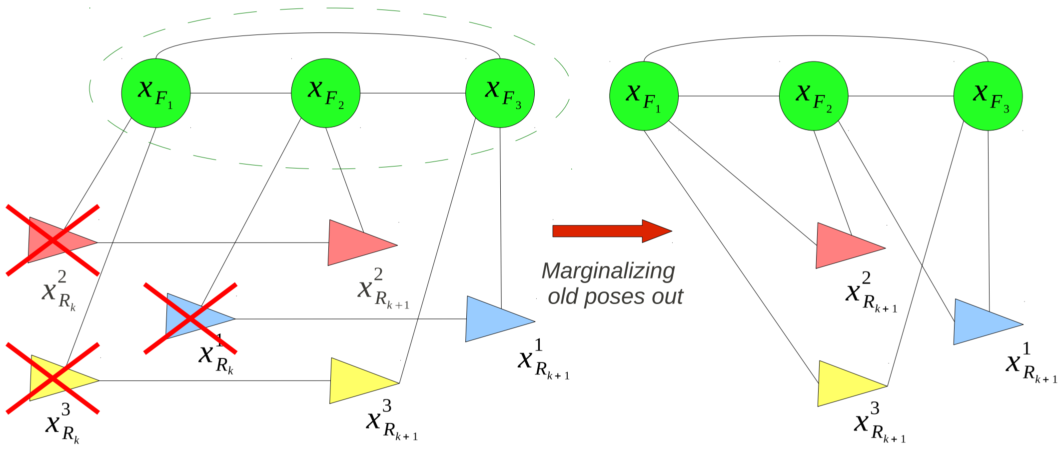

Fig. 3. GMRF of the robot formation. The example shows three robots that make observations of some map features during two consecutive steps. Robots are only related through the features, i.e. they are conditionally independent given the map (green nodes). Previous robot positions but the first one $\mathbf{x}^j_{R_k}$ are marginalized out (left). Red, blue and yellow nodes are separated by green nodes and therefore they are CI as in Fig. 1.

Algorithm 3 **sendMessages**$(\mathbf{I}^j_{s+p}, \mathbf{i}^j_{s+p})$

$\mathbf{I}^{m,j}_{F,s}, \mathbf{i}^{m,j}_{F,s}$ {*Feature Marginal of robot j stored from last synchronization at instant s*}

$(\mathbf{I}^{m,j}_{F,s+p}, \mathbf{i}^{m,j}_{F,s+p}) = \texttt{marginalizedOut}(\mathbf{x}_{R_{s+p}}, \mathbf{I}^j_{s+p}, \mathbf{i}^j_{s+p}))$

$$\begin{aligned} \mathbf{I}^{new,j}_{F,s+1:s+p} &= \mathbf{I}^{m,j}_{F,s+p} - \mathbf{I}^{m,j}_{F,s} \\ \mathbf{i}^{new,j}_{F,s+1:s+p} &= \mathbf{i}^{m,j}_{F,s+p} - \mathbf{i}^{m,j}_{F,s} \end{aligned} \tag{13}$$

$\texttt{broadcast}(\mathbf{I}^{new,j}_{F,s+1:s+p}, \mathbf{i}^{new,j}_{F,s+1:s+p})$

Algorithm 4 **receiveMessages**$()$

for $r \neq j$ **do**
  $\mathbf{I}^{r,m}_{F,s+p}$, $\mathbf{i}^{r,m}_{F,s+p}$ {*Feature Marginal received from robot r*}
  $\mathbf{I}^j_{s+p} = \mathbf{I}^j_{s+p} \boxplus \mathbf{I}^{r,m}_{F,s+p}$
  $\mathbf{i}^j_{s+p} = \mathbf{i}^j_{s+p} \boxplus \mathbf{i}^{r,m}_{F,s+p}$
end for

6. DISTRIBUTED LOCALIZATION AND MAPPING ALGORITHM FOR ROBOT FORMATIONS

In this section we combine for each robot the techniques described in sections 4 and 5. On the one hand, the formation works in a local region of the map to reduce the computational cost. As the robots navigate relatively close in the formation they are localized in the same submap. On the other hand, each robot maintains its own estimation and is in charge of its own observations to update the map. From time to time, the robots get synchronized to obtain the same map estimate. At these instants, each robot also broadcasts its best estimated position to maintain the formation structure. Notice that, after a synchronization, the estimated pose of the formation and the map coincide with that of a centralized version.

Our distributed method for each member of the robot formation is presented in Algorithm 5. First, the robot is localized in a local working region (GlobalToLocal). While it remains in the same region ($mapChange = false$), the standard EIF operations are carried out to estimate its position and features location but without marginalizing out the initial position of the robot in the submap (LocalEIF). Based on the knowledge about the formation structure and using the spring-damper approach described in Urcola et al. (2008) as navigation strategy, each robot is able to predict the position of the rest so that the robots do not stop navigating in formation towards their goal in absence of communication. If a synchronization event *check_sync* is registered (e.g. after a determined period of time, or when a robot makes a request, etc.), an exchange of messages takes place to update the states of the team with the same information. Also, when the formation changes to a new local region ($mapChange = true$), the robots synchronize to update its total map and another iteration of the main execution loop is realized. The new information that each robot receives about the features will affect its own position estimation. For this reason and to update their knowledge about the formation, each time there is an exchange of messages, each robot sends its improved position to the rest of the robots (sendPositionToFormation) and receives the information from the rest (receivePositionsFromFormation).

6.1 Computational and Communication Complexity

While each robot is working in its own local submap, it performs the LocalEIF algorithm. The computational cost of this filter depends on the number of features in the submap. Since we are not adding new features to the map, the number of features remains constant and the cost will be ($\mathcal{O}(1)$) while working in the local region.

When a change of submap is carried out, there are two operations involved, LocalToGlobal, to update the global map, and GlobalToLocal, to extract a new region of the map. As it was explained in section 4, the operation LocalToGlobal consists in adding the new information to the global map, therefore, its cost is $\mathcal{O}(1)$. In GlobalToLocal, we find the most costly operation of Algorithm 5, where we have to invert almost the whole map, (Eq. 6), leading to a

Algorithm 5

$mapChange = false$
while $(k < nsteps)$ **do**
  $(\mathbf{i}^m_{l,k}, \mathbf{I}^m_{l,k})$=`GlobalToLocal`$(\mathbf{i}_k, \mathbf{I}_k)$
  while not $mapChange$ **and** $(k < nsteps)$ **do**
    $(\mathbf{i}^m_{l,k+1}, \mathbf{I}^m_{l,k+1}, mapChange)$=`LocalEIF`$(\mathbf{i}^m_{l,k}, \mathbf{I}^m_{l,k})$
    if $mapChange$ **or** $checkSinc()$ **then**
      `sendMessages`$(\mathbf{i}^m_{l,k+1}, \mathbf{I}^m_{l,k+1})$
      `receiveMessages()`
      `sendPositionToFormation()`
      `receivePositionsFromFormation()`
    end if
    $k = k + 1$
  end while

  {*After p local EIF steps, a global update takes place*}

  $(\mathbf{I}_{k+p}, \mathbf{i}_{k+p})$=`LocalToGlobal`$(\mathbf{I}^m_{l,k}, \mathbf{i}^m_{l,k}, \mathbf{I}^m_{l,k+p}, \mathbf{i}^m_{l,k+p}, \mathbf{I}_k, \mathbf{i}_k)$
end while

cost of $\mathcal{O}(n^3)$ in the worst case, being n the fixed number of features of the prior map. Unlike other algorithms, this operation only takes place at each map changing step. As it was studied in Paz and Neira (2006), there exists a trade-off between the size of the local maps and the frequency with which the robots need to change of submaps. If the submaps are small, global updates will be more frequent, but, on the other hand, larger submaps will increase the computational cost of local updates.

Concerning the communication, the total amount of information to be sent to other robots is bounded to the number of features in the local map. Since each robot broadcasts the new information added to the submap to the rest, the communication complexity scales with the number of robots $\mathcal{O}(n_r)$.

7. RESULTS

Through the following simulation results we want to show the advantages of the distributed submapping algorithm proposed in this paper. We have designed a simulation environment of 30x30m where three robots set in a triangle formation have to navigate along a 120m loop scenario (Fig. 4). Each robot has a prior stochastic map of the navigation area divided in submaps of 10x10m. The current local region $\mathbf{x}_{F_l}$ is common to all the robots and is selected depending on which features are being observed by the robots, thus, this local region can be composed of several submaps.

One of the main advantages of this algorithm is the improvement of the given stochastic map where uncertainties and errors of the map features decrease. This can be seen in the zoomed area of Fig. 4, where the previous map and the final map are depicted. This improvement not only affects the covariances but also the error obtained in each of the features as shown in Fig. 5.

As a direct consequence of the map improvement, the robots are localized more accurately. Figure 6 shows the Root Mean Squared Error obtained on each component of the localization of the robot formation.

The next advantage concerns the computational cost. Three different implementations have been compared.

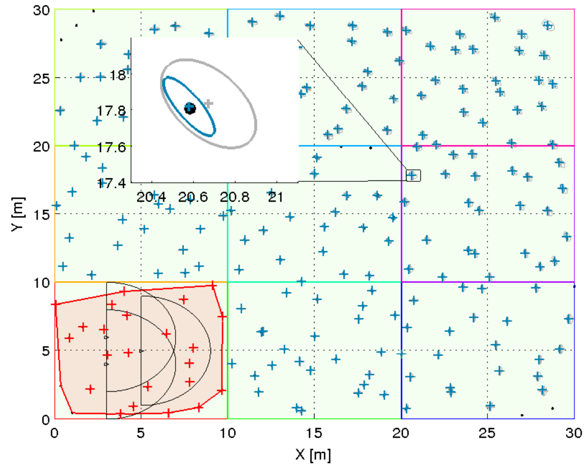

Fig. 4. Stochastic map of the simulation environment and initial setting of the formation, where the current local region is depicted in red. Reduction of the covariances (blue) with respect the a priori map (grey). Black dot in the zoomed in area represents the ground truth of the feature.

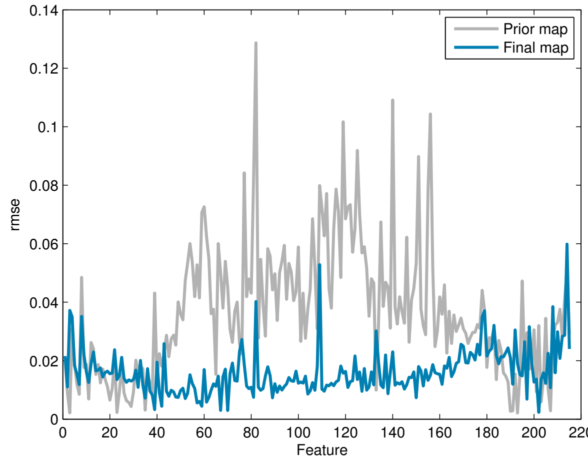

Fig. 5. Comparison between the RMSE per feature in the a priori map and in the final map. In the a priori map, maximum errors are due to far features w.r.t. the base reference while the minimum errors are due to features involved during a loop closure event or those that are near to the origin. After running the distributed algorithm, the error is more balanced in the whole final map.

First, the submapping and distributed technique proposed in this paper (Dist+sub) where each robot updates its own local and global maps based on its own observations and synchronizes with the other robots to obtain a better estimation. Second, a centralized version of the submapping technique (Cent+sub) based on Piniés and Tardós (2008), where the robot leader is the one who updates the local and global maps using the observations gathered by all the robots. Finally, a centralized EKF-based version in which the leader does not work with local maps but only with the global map (Cent+glob).

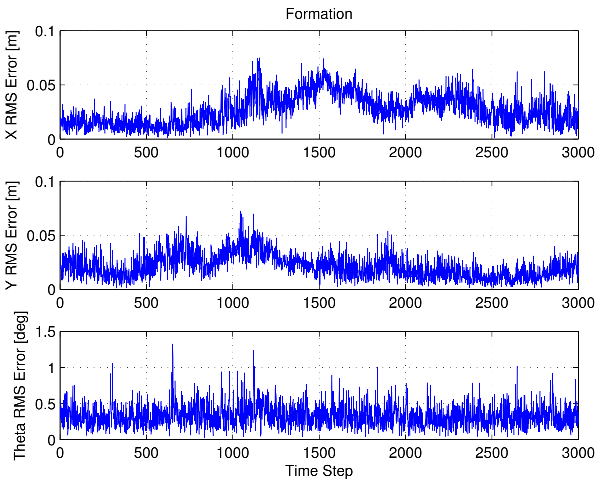

Fig. 6. RMSE error of all the robots of the formation. The average error in the X and Y components is around 2cm, while the orientation error is around 0.4 degrees.

Figure 7 shows a comparison between the first (Dist+sub) and second version (Cent+sub). We have supposed the worst case time complexity for the first implementation in which the formation synchronizes at each time step and an additional time per step appears due to the execution of the send-receive operations. However, the cost of the map update in the distributed version (green line) is lower, resulting in a less time complexity with respect to the centralized implementation (red line). Peaks in the times are due to global updates, where Algorithms 1 and 2 are executed.

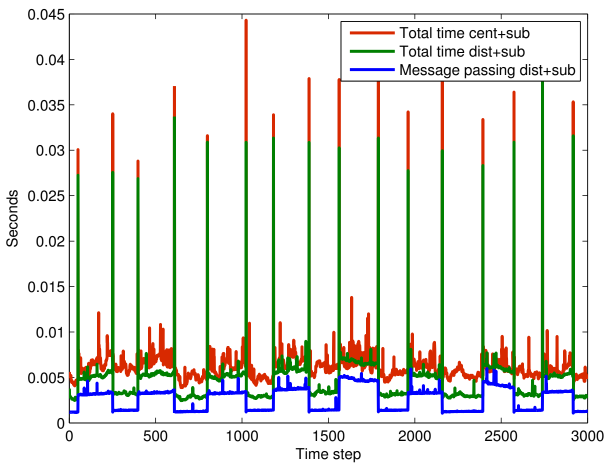

Fig. 7. Times per step for the centralized and distributed submapping implementations. Worst case where robots synchronize at each step is shown. For that reason, synchronization time (blue), which affects to the distributed implementation, is never zero.

In Fig. 8, the computational cost of the first (Dist+sub) and the third version (Cent+glob) is depicted. We can see how even when a global update occurs, the times of the distributed implementation are lower than the global version. We can also observe that, since we are not adding new features to the map, the cost of the global EKF mapping remains approximately constant. In the distributed implementation, the times are also constant while working in the same submap, but these times change depending on the number of features of the submaps (e.g. time at instant 2250 and at instant 2500). Note that, in a real experiment, an additional time would have to be considered in both cases, due to the data association process. In the global centralized EKF mapping, the leader would perform this task by matching all the observations from all robots, thus, this time would be proportional to the number of robots and observations whereas in the distributed version, this time would only depend on the number of observations.

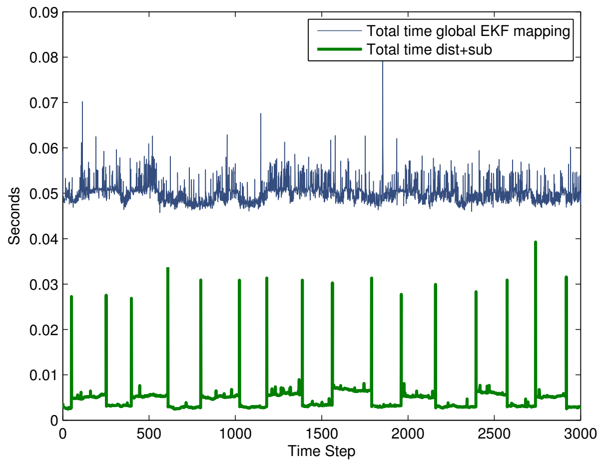

Fig. 8. Time comparison between the global centralized and the submapping distributed implementations. Synchronization messages are transmitted at every step.

Finally, we want to emphasize that the result obtained with the distributed submapping method proposed in this paper is equivalent to the result obtained in a centralized-global mapping implementation. Each time the robots perform a synchronization, they will obtain the optimal solution (i.e. equivalent to the centralized). In Fig. 9 we show the consistency ratio NEES/$\chi^2_{r,1-\alpha}$ (Bar-Shalom et al. (2001)), $r = \dim(\mathbf{x}_{F_l})$ of the local map features when robots synchronize every 10 time steps. We can see how, while there is no synchronization, NEES for each robot's features is different, but when a synchronization occurs the solution obtained is equal to the centralized. Besides, the estimation of the map features is consistent, since the ratio NEES/$\chi^2_{r,1-\alpha} \leq 1$.

8. CONCLUSIONS AND FUTURE WORK

In this work we have proposed a distributed estimation algorithm within the framework of robot formations for long term performance. Unlike other distributed data fusion systems, we use a prior map such that our method can efficiently tackle the localization of the robot formation at the same time the map is dynamically improved with new observations. The algorithm proposed does not rely on a central server improving flexibility and robustness. This is achieved by describing the distributed estimation problem as a GMRF, which allows us to take double advantage of the CI properties to reduce the computational and communication requirements: as first result the formation

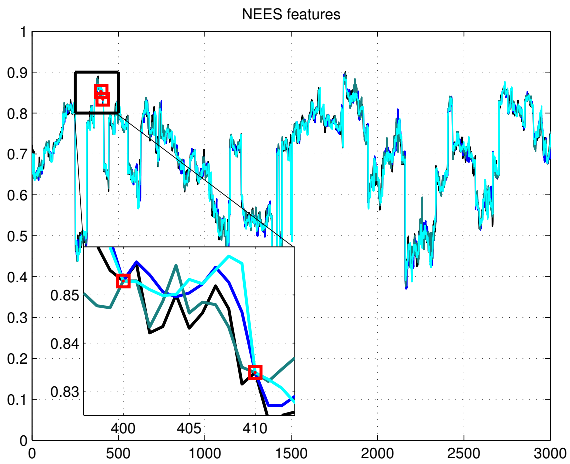

Fig. 9. Consistency ratio NEES/$\chi^2_{r,1-\alpha}$ of the features in the centralized (black) and the distributed (the three blue lines, one for each robot of the formation) implementations. In the zoomed area, red boxes correspond to synchronization times (every 10 time steps) where centralized and distributed results are exactly the same. If synchronizations occurred at every time step, the three blue lines would coincide with the black line.

only experience constant updates whereas global updates are postponed until a new local region transition takes place; a second cost reduction is achieved due to the fact that robots only get indirectly related through the observation of common map features. In consequence, the load in the communication channel will only scale linearly with the number of resources since each robot broadcasts only an information summary of features observed from the last robot formation communication.The algorithm results show an accuracy improvement of the a priori map, being the final result equivalent to the one obtained in a centralized implementation with a lower computational effort.

Our distributed algorithm could deal with asynchronous communications between robots by adopting the same strategy described in Grime and Durrant-Whyte (1994). This will be part of a future implementation in real scenarios. In addition we will consider an adaptive feature management policy to identify persistent elements and cope with map structural changes encountered in the environment during long term performance. A further step in the estimation process would be to extend the conditional properties to a distributed non linear optimization algorithm. This will be part of our immediate research.

REFERENCES

Bailey, T., Bryson, M., Mu, H., Vial, J., McCalman, L., and Durrant-Whyte, H. (2011). Decentralised cooperative localisation for heterogeneous teams of mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2859–2865.

Bar-Shalom, Y., Li, X.R., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*. Wiley and Sons.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer, second edition.

Cunningham, A., Paluri, M., and Dellaert, F. (2010). DDF-SAM: Fully distributed SLAM using constrained factor graphs. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 3025–3030. Taipei, Taiwan.

Eustice, R., Singh, H., and Leonard, J. (2006). Exactly sparse delayed-state filters for view-based SLAM. *Robotics, IEEE Transactions on*, 22(6), 1100–1114. doi: 10.1109/TRO.2006.886264.

Grime, S. and Durrant-Whyte, H. (1994). Data fusion in decentralized sensor networks. In *Control Engineering Practice*, volume 2, 849–863.

Lázaro, M.T. and Castellanos, J.A. (2010). Localization of probabilistic robot formations in SLAM. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 3179–3184. Anchorage, Alaska.

Leung, K., Barfoot, T., and Liu, H. (2011). Distributed and decentralized cooperative simultaneous localization and mapping for dynamic and sparse robot networks. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 3841–3847. Shangai, China.

Nerurkar, E.D., Roumeliotis, S.I., and Martinelli, A. (2009). Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1402–1409. Kobe, Japan.

Nettleton, E., Thrun, S., Durrant-Whyte, H., and Sukkarieh, S. (2003). Decentralised SLAM with low-bandwidth communication for teams of vehicles. In *Proc. Int. Conf. on Field and Service Robotics*, 179–188. Lake Yamanaka, Japan.

Paz, L.M. and Neira, J. (2006). Optimal local map size for ekf-based slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*. Beijing, China.

Piniés, P. and Tardós, J.D. (2008). Large Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision. *Transactions on Robotics*, 24(5), 1094–1106.

Roumeliotis, S.I. and Bekey, G.A. (2002). Distributed multi-robot localization. *IEEE Trans. on Robotics and Automation*, 18(5), 781–795.

Urcola, P. and Montano, L. (2011). Adapting robot team behavior from interaction with a group of people. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2887–2894. San Francisco, CA, USA.

Urcola, P., Riazuelo, L., Lázaro, M.T., and Montano, L. (2008). Cooperative navigation using environment compliant robot formations. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2789–2794. Nice, France.

Walter, M.R., Eustice, R.M., and Leonard, J.J. (2007). Exactly sparse extended information filters for feature-based SLAM. *The International Journal of Robotics Research*, 26(4), 335–359.