

# Range-Only SLAM for Robot-Sensor Network Cooperation

A. Torres-González · J. R. Martínez-de Dios · A. Ollero

Received: date / Accepted: date

**Abstract** This work is motivated by schemes of robot-sensor network cooperation where sensor nodes –beacons– are used as landmarks for Range-Only (RO) Simultaneous Localization and Mapping (SLAM). In most existing RO-SLAM techniques beacons are considered as passive devices ignoring the capabilities they are actually endowed with. This paper proposes a RO-SLAM scheme that distributes the measurements gathering and integration between the beacons surrounding the robot. It naturally integrates inter-beacon measurements, significantly improving map and robot estimations and speeding up beacon initialization. The proposed scheme is based on Sparse Extended Information Filter (SEIF) and it is proven that it preserves the constant time and sparsity properties of SEIF and thus, inherits its efficiency and scalability. As a result, our scheme has lower robot and map estimation errors, faster beacon initialization and lower computer requirements than existing methods. This paper experimentally validates and evaluates the proposed method for 3D SLAM using an octocopter.

**Keywords** SLAM · Sensor Networks · Range-Only SLAM · UAS

---

This work has been supported by EU Project AEROARMS Ref. H2020-ICT-2014-1-644271 and the AEROMAIN project funded by the Spanish R&D plan (DPI2014-59383-C2-1-R). A. Torres-González thanks the Ministerio de Educación y Deportes FPU Program. J.R. Martínez-de Dios acknowledges EU Project EUROCC funded under contract 608849.

---

A. Torres-González (✉) · J. R. Martínez-de Dios · A. Ollero  
Robotics, Vision and Control Research Group, University of Seville,  
Camino de los Descubrimientos s/n, 41092 Seville, Spain.  
Tel.: +34-954-487349  
Fax: +34-954-487340  
E-mail: [arturotorres@us.es](mailto:arturotorres@us.es), [jdedios@us.es](mailto:jdedios@us.es), [aollero@us.es](mailto:aollero@us.es)

## 1 Introduction

This paper deals with Range-Only (RO) Simultaneous Localization and Mapping (SLAM) in which the robot integrates range measurements to radio beacons deployed in the scenario in order to build a map of an unknown environment and to self-localize in that map. Our scheme uses the nodes of a sensor network as beacons –landmarks– for RO-SLAM. In short, sensor networks consist of spatially distributed autonomous devices –nodes– employed to gather measurements of an environment or a process and to cooperatively transmit them to a main location. Visual SLAM does not require any deployment in the environment, which is critical in many problems, and a very high number of very good visual SLAM solutions have been developed [22]. However, RO-SLAM using radio beacons is more interesting than visual SLAM in many other cases. RO-SLAM is independent of lighting conditions and does not require line-of-sight operation, whereas visual SLAM is often sensitive to changing lighting conditions and is not suitable in presence of smoke, heavy dust [3] or specular reflections. Besides, using radio beacons as landmarks naturally solves the data association problem, typical of visual SLAM.

Consider a GPS-denied scenario where a large number of sensor nodes have been deployed at unknown locations. For instance, they have been placed at random locations for real-time monitoring of an accident, or they are used for monitoring an industrial facility and their exact location was not registered during deployment. Each sensor node gathers measurements and transmits them to a Monitoring Station. We assume that each node can measure the distance to other nodes. This is not a hard requirement, in fact most Commercial Off-The-Shelf (COTS) sensor network nodes can measure the radio signal strength (RSSI) of incoming packets and estimate the range to the emitting node [1]. Accurate mapping and robot localization is critical for the navigation of the

robot in complex GPS-denied environments. Also, the localization of sensor nodes –mapping– is necessary to geolocate the measurements collected. Besides, knowing the location of robots and nodes enable robot-sensor node cooperative missions of interest in these scenarios such as sensor node transportation and deployment [5, 16] or collection of data from sensors [6]. The potentialities of robot-sensor network collaboration has originated significant interest in RO-SLAM techniques that employ sensor nodes as beacons, see e.g. [4, 18, 19]. However, most reported techniques consider beacons as passive landmarks and do not exploit the capacities they are actually endowed with.

This paper proposes a RO-SLAM scheme that exploits the robot-sensor network collaboration by distributing measurement gathering and integration in SLAM between the beacons around the robot. It is based on Sparse Extended Information Filter (SEIF) [25], and inherits its efficiency and scalability. Its distributed approach shares resource consumption, reducing robot CPU burden, and at the same time naturally integrates inter-beacon measurements (measurements between two beacons), which improves map and robot localization accuracies. It is proven in the paper that the proposed scheme accumulates higher amount of information than the conventional SEIF SLAM and at the same time, it preserves the sparsity of the information matrix and its constant time property. The paper presents the scheme, evaluates and compares its performance with existing methods in experiments performed with an octorotor aerial robot.

This paper is an extended version of our work [26]. The main improvements of this paper over [26] are:

- a new section that justifies the validity of the proposed approach analyzing its effect on the building of the information matrix. First, it theoretically proves the preservation of the main properties of SEIF using common assumptions. Then, it analyzes with simulations their validity when the assumptions are relaxed;
- extension to 3D SLAM and integration of the scheme with an octorotor aerial robot;
- a new section with the extensive validation of the proposed method in 3D SLAM using an octorotor.
- extension and more detailed related work. Also, the paper has been restructured and all sections have been completed and rewritten for clarity.

The paper is organized as follows. Section 2 summarizes the main existing RO-SLAM methods. Section 2.1 briefly summarizes SEIF SLAM as an introduction to our method, which is detailed in Section 3. Section 4 analyzes the building of the information matrix in proposed scheme and proofs that it preserves the main properties of SEIF SLAM. Section 5 evaluates and compares with existing methods its performance and robustness in 3D SLAM in real experiments and simulations. The conclusions are in the last section.

## 2 Related work

A good number of RO-SLAM methods have been developed in the last years. Most of them employ approaches based on Extended Kalman Filter (EKF) SLAM or on Fast-SLAM. FastSLAM [2] factorizes the state vector dividing it in the vehicle pose estimation and the map estimation. It is more flexible than EKF due to the use of a Particle Filter as the core of the algorithm, which allows having different noise distributions apart from only Gaussian as in the case of EKF. In FastSLAM each particle of the filter represents an hypothesis of the state (robot pose and map). EKF SLAM [8, 17] is a well-known method, obtains good results in most implementations but it lacks scalability.

Different strategies have been developed in the past years to improve efficiency and scalability in SLAM. The majority of them use the canonical form of the Gaussian distribution. Thin Junction Tree Filters [21] employ an special data structure –called junction tree– to represent the information matrix. Work [21] proposes approximations to keep these junction trees simple –thin– regardless of the map size. Treemap Filters [12] hierarchically divide the map into local regions.

Dual to Kalman Filters, Information Filters (IFs) represent the state by its canonical form based on the information vector and the information matrix. In Extended Information Filter (EIF) SLAM the update stage is very efficient –additive– but the prediction stage requires operations with the whole information matrix, involving bad scalability with the map size. Besides, it is a full SLAM solution, i.e. it solves the SLAM problem after all information has been gathered, not being suitable for on-line applications.

Sparse Extended Information Filter (SEIF) [25, 15] solves on-line SLAM maintaining a sparse representation of the information matrix, which simplifies the matrix operations, improving efficiency and scalability. It has been demonstrated in [25, 11] that many of the off-diagonal elements of the information matrix are relatively close to zero. SEIF SLAM enforces the sparsity of the information matrix, enabling efficient algorithms for motion update and state recovery. As a result, it can be executed in constant time regardless of the map size. SEIF for feature-based SLAM has been researched in some works. In [9] a modification for ensuring the consistency of the global map estimate was proposed. SEIF SLAM for multi-robot systems was used in [24].

Most RO-SLAM techniques employ only robot-beacon range measurements. Work [8] was the first in using also measurements between beacons (inter-beacon). Integrating inter-beacon measurements improves map and robot estimation accuracies and speeds up beacon initialization. Despite its advantages few methods employ them since they involve significantly higher consumption of resources including the computational power required to integrate the higher number of measurements. In a previous work [28] we addressed

the increment in resource consumption proposing a centralized EKF SLAM scheme that dynamically disconnects the integration of inter-beacon measurements in SLAM when the desired uncertainties have been met. In this paper we propose an efficient robot-beacons distributed SEIF scheme where beacons actively participate in gathering and integrating measurements in SLAM, sharing resource consumption and ensuring constant time execution. Work [7] presented a decentralized scheme to self-localize a sensor network with the help of a mobile robot. It is based on Loopy Belief Propagation (LBP) [13], so it requires the sensor network to be sparse –not fully connected– which is an important limitation in many applications. In case of having loops in the network some information can be counted twice, making it more difficult to accurately recover the exact state representation.

## 2.1 Range Only SEIF SLAM in a nutshell

This section briefly summarizes the SEIF SLAM algorithm as an introduction to the proposed scheme. Most expressions have been omitted. For notation simplicity, time subindex  $t$  has been also omitted. Refer to [23] for further details. The adopted state vector  $\mathbf{x}$  is comprised of the robot position ( $x_r$ ) and the location of all the beacons in the map ( $x_1, x_2, \dots, x_N$ ). SEIF SLAM is based on Extended Information Filter (EIF). Dual to EKF, EIF represents the state vector by the information vector  $\xi = \Sigma^{-1}\mu$  and the information matrix  $\Omega = \Sigma^{-1}$ , where  $\mu$  is the mean of the state vector and  $\Sigma$  is the covariance matrix.  $\Omega$ , see (1), is symmetric and positive-semidefinite. Each off-diagonal entry of  $\Omega$  –called *link* [25]– represents the relation between two elements in  $\mathbf{x}$ .

$$\Omega = \begin{pmatrix} \Omega_{r,r} & \Omega_{r,1} & \cdots & \Omega_{r,N} \\ \Omega_{1,r} & \Omega_{1,1} & \cdots & \Omega_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_{N,r} & \Omega_{N,1} & \cdots & \Omega_{N,N} \end{pmatrix} \quad (1)$$

At any time in the SLAM operation some of the off-diagonal elements of  $\Omega$  are zero meaning lack of information between the involved elements; some of them have high values –strong links– meaning high information; and a number of them have values close to zero –weak links. Weak links have much lower impact on the estimation than strong links but both involve similar computational burden. SEIF maintains a sparse representation of  $\Omega$  by keeping the number of active beacons –beacons with non-zero links to the robot– bounded by a threshold. Every time the number of active beacons is above the bound, the sparsification step is performed deactivating the beacons with the weakest links.

Measurement update in information filters modifies only the entries of  $\Omega$  corresponding to the elements involved in the measurement. Factorizing  $\Omega$  allows efficient update stage

regardless of the map size. Also, the sparsity of  $\Omega$  significantly reduces the computational burden required in SEIF for the prediction stage. For linearizing the prediction and observation models it is required to recover the state estimate  $\mu$  from the predicted  $\bar{\Omega}$  and  $\bar{\xi}$ . The whole state is not needed, only the states of the robot and active beacons are required. Of course, enforcing sparseness in  $\Omega$  involves an approximation error in the estimations obtained by SEIF. Work [23] suggests using sparsification bounds in the range [4 – 10] in order to balance accuracy degradation and burden reductions.

We could not find any paper that used SEIF SLAM with range measurements. The observation model we adopted for the integration of robot-beacon range measurement  $z_{r,j}$  taken by the robot to beacon  $b_i$  is:

$$h_{r,i}(\mu) = \sqrt{\delta_x^2 + \delta_y^2 + \delta_z^2}, \quad (2)$$

where  $\delta_x = \mu_x - \mu_x^i$ ,  $\delta_y = \mu_y - \mu_y^i$  and  $\delta_z = \mu_z - \mu_z^i$ , where  $[\mu_x, \mu_y, \mu_z]$  and  $[\mu_x^i, \mu_y^i, \mu_z^i]$  are respectively the estimated positions of the robot and  $b_i$ .  $h_{r,i}$  is nonlinear. Its Jacobian is:

$$H_{r,i} = \frac{\partial h_{r,i}}{\partial \mu} = \begin{bmatrix} \frac{\delta_x}{h_{r,i}} & \frac{\delta_y}{h_{r,i}} & \frac{\delta_z}{h_{r,i}} & \cdots & -\frac{\delta_x}{h_{r,i}} & -\frac{\delta_y}{h_{r,i}} & -\frac{\delta_z}{h_{r,i}} & \cdots \end{bmatrix} \quad (3)$$

All the entries of  $H_{r,i}$  that are not shown in (3) are zero. All the entries are zero except for those corresponding to the robot and to  $b_i$ . Our method also integrates inter-beacon measurements, such as  $z_{i,j}$  gathered by beacon  $b_i$  to  $b_j$ . The model adopted  $h_{i,j}$  is an extension of that in (2). Again, all the entries of Jacobian  $H_{i,j}$  are zero except for those corresponding to the beacons involved in the measurement:

$$H_{i,j} = \begin{bmatrix} 0 & \cdots & \frac{\delta_x}{h_{i,j}} & \frac{\delta_y}{h_{i,j}} & \frac{\delta_z}{h_{i,j}} & \cdots & -\frac{\delta_x}{h_{i,j}} & -\frac{\delta_y}{h_{i,j}} & -\frac{\delta_z}{h_{i,j}} & \cdots \end{bmatrix} \quad (4)$$

Range measurements have the problem of partial observability. The SEIF SLAM filter should be combined with an auxiliary tool for beacon initialization such as are Particle Filters [17], Probability Grids [8,20] and trilateration methods [18]. Trilateration methods, although simple and efficient, are very sensitive to measurement noise and can lead to high initialization errors. Probability Grids provide better initialization but their accuracy depends on the size and resolution of the grid. Particle Filters (PFs) are the most widely applied method. They provide better accuracy and various mechanisms have been developed to reduce their computational burden. We adopted PFs in the experiments shown in the paper. When the robot takes the first measurement from beacon  $b_i$ , the initialization of  $b_i$  is started and then beacon  $b_i$  enters at the “initialization phase”. It keeps in this stage until the auxiliary tool converges and an initial estimation of the location of  $b_i$  is computed. Then, it is added to the state vector. From now on we say that  $b_i$  is at the “state vector phase”.

### 3 Robot-sensor network distributed SEIF SLAM

The proposed work is strongly motivated by our envisioned application: aerial robot autonomous navigation for maintenance and repairing in industrial plants, which are environments with poor GPS reception. Figure 1 shows a picture of a typical environment in our problem. These scenarios are not well textured, they contain many pipes, which often have few features. Also, pipes frequently originate specular reflections and shadows causing drastic local lighting changes involving severe problems in visual and RGB-D SLAM techniques. Although very good visual and RGB-D SLAM solutions have been developed, in these environments they are not always robust enough for safe robot localization and navigation. In our problem RO-SLAM techniques that use sensor nodes—beacons—as landmarks are an interesting alternative. Each beacon is endowed with sensing, computing and communication capabilities and can measure the distance to the robot or to other beacons within its sensing region. These are not constraints in practice. A wide variety of Commercial Off-The-Shelf sensor nodes fulfill them. Our method performs satisfactorily with low densities of beacons deployed without special care in the scenario. Besides, in many industrial plants sensor nodes are already deployed and currently used for process monitoring & maintenance.

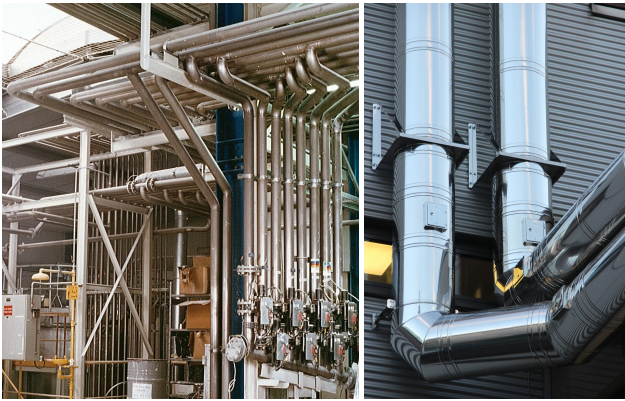


Fig. 1 Pictures of the objective scenario of the proposed method.

In conventional RO-SLAM techniques the robot gathers range measurements to the beacons within its sensing region and integrates these measurements in the update stage. In the proposed scheme measurement gathering and integration in SLAM is distributed among the beacons. The beacons involved at time  $t$  can be classified into two sets: direct beacons and indirect beacons. Direct beacons are those that at time  $t$  are within the robot sensing region ( $SR_r$ ). The set of direct beacons at time  $t$  is  $DBS_t = \{b_i : b_i \in SR_r\}$ . Indirect beacons are those that are outside  $SR_r$  but within the sensing region of a direct beacon. The set of indirect beacons at  $t$  is:

$$IBS_t = \{b_j : b_j \notin SR_r, b_j \in SR_i \forall b_i \in SR_r\}, \quad (5)$$

where  $SR_i$  is the sensing region of beacon  $b_i$ .

The operation of the presented scheme is as follows. The SLAM prediction stage is executed in the robot. The robot keeps  $LNB$ , the list with the direct beacons it has detected. Then, the robot broadcasts an *UpdateReq* message that includes  $LNB$  and the predicted state. Each direct beacon  $b_i$  receives the message and extracts  $LNB$ . If  $b_i$  finds itself in  $LNB$ , it gathers a range measurement to the robot ( $z_{i,r}$ ) and to each of the beacons within its sensing region  $SR_i$ . This measurement set is designed as  $MS_i = \{z_{i,r}, z_{i,j} \forall b_j \in SR_i\}$ . Then, each direct beacon  $b_i$  computes with the measurements in  $MS_i$  its contribution to the update stage and transmits it to the robot in an *UpdateResp* message. The update stage in SLAM represented in the information form is additive. Thus, the robot reconstructs the SLAM updated state by adding the contributions it received. Figure 2-top summarizes the main tasks and message interchanged in the proposed scheme..

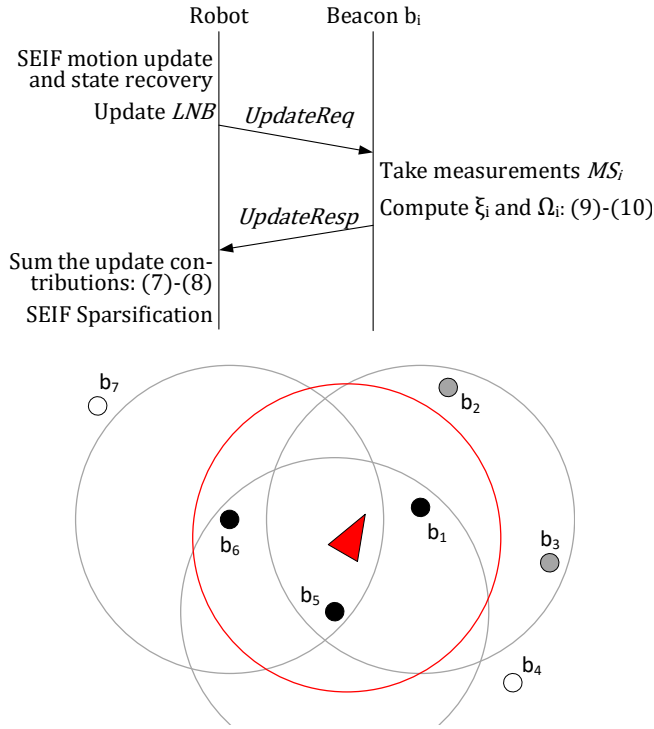
Measurement gathering is illustrated in Fig. 2-bottom. The red and gray circumferences represent the sensing regions of the robot and beacons, respectively. Direct beacons are represented in black color and, indirect in gray:  $DBS_t = \{b_1, b_5, b_6\}$  and  $IBS_t = \{b_2, b_3\}$ . In conventional RO-SLAM the robot gathers and integrates measurements to direct beacons,  $\{z_{r,1}, z_{r,5}, z_{r,6}\}$  in Fig. 2. In our scheme each direct beacon  $b_i$  gathers and integrates the measurements in its  $MS_i$ , e.g.  $b_5$  gathers  $MS_5 = \{z_{5,r}, z_{5,1}, z_{5,6}\}$ . Thus, when the robot completes the SLAM update stage it has integrated measurements  $\{z_{1,r}, z_{1,2}, z_{1,3}, z_{1,5}, z_{5,r}, z_{5,1}, z_{5,6}, z_{6,r}, z_{6,5}\}$ .

Our scheme integrates all robot-beacon and inter-beacon measurements that involve one or more direct beacons, naturally avoiding repeated measurements. It gathers one measurement for all these distances except for those between two direct beacons, e.g.  $b_1$  and  $b_5$ . In these cases two different measurements are gathered, e.g.  $z_{1,5}$  and  $z_{5,1}$ . The method is also robust to message loss, as is shown in Section 5.

#### 3.1 Operation of the robot

The operation of the robot in one prediction-update cycle is summarized in Alg. 1. First, the robot computes the SLAM prediction stage. We assume static beacons and a robot with nonlinear kinematic model. Thus, its Jacobian should be computed at each time, which requires recovering  $\mu$ . Our scheme uses the efficient algorithm described in [23] for motion update and state recovery (step 1 in Alg. 1). This algorithm computes the predicted information vector  $\bar{\xi}$ , the information matrix  $\bar{\Omega}$  and also the recovered predicted estimate  $\mu$ .

Next, the robot broadcasts an *UpdateReq* message that includes the predicted estimate  $\mu$ . Transmitting the whole



**Fig. 2** (Top) Main tasks and messages interchanged between the robot and beacon  $b_i$  in the proposed scheme. (Bottom) Measurement gathering in a conventional RO-SLAM and in the proposed scheme. The red triangle represents the robot. Direct beacons are represented with black circles and indirect, with gray circles. Circumferences represent the sensing regions of the robot (red) and of beacons (gray).

### Algorithm 1 Summary of the operation of the robot

**Require:**  $\xi_{t-1}, \Omega_{t-1}$

- 1: SEIF SLAM motion update and state recovery
- 2: Update  $LNB$  and create  $UpdateReq$  message
- 3: Broadcast  $UpdateReq$  message
- 4: Receive  $UpdateResp$  messages from beacons
- 5: Sum SLAM update contributions to  $\bar{\xi}$  and  $\bar{\Omega}$  as in (7) and (8)
- 6: SEIF SLAM Sparsification
- 7: **return**  $\xi_t, \Omega_t$

$\mu$  is not suitable in cases with large numbers of beacons: it would require broadcasting large –or several– messages, increasing message losses. Besides, the greater part of  $\mu$  is not of interest for direct beacons. Only the elements in  $\mu$  required for each direct beacon in  $LNB$  are transmitted. Each direct beacon  $b_i$  gathers range measurements to the robot and to the beacons  $b_j \in SR_i$ . For integrating them it needs  $\mu_i, \mu_r$  and  $\mu_j \forall b_j \in SR_i$ .  $ev_i$  is the vector with the estimates required for direct beacon  $b_i$  to perform its SLAM update:

$$ev_i = [\mu_r \ \mu_i \ \mu_j]^T \quad (6)$$

At the beginning  $LNB$  is assumed empty. If a direct beacon that received the  $UpdateReq$  message does not find itself in  $LNB$ , it sends a  $BeaconDiscovery$  message to the robot

with its ID and the IDs of the beacons that are within its sensing region. Then, the robot will add it to  $LNB$ . When the robot does not receive update contributions from beacon  $b_i$  within  $LNB$  in a number of consecutive times, it is interpreted that  $b_i$  is currently outside  $SR_r$  and is deleted from  $LNB$ .

$UpdateResp$  message from  $b_i$  contains its contribution to the SLAM update ( $\xi_i$  and  $\Omega_i$ ). The robot receives the  $UpdateResp$  messages and when a timeout expires, it reconstructs the updated state ( $\xi$  and  $\Omega$ ) adding the predicted  $\bar{\xi}$  and  $\bar{\Omega}$  to the contributions it received:

$$\xi = \bar{\xi} + \sum_i F_i^T \xi_i, \quad (7)$$

$$\Omega = \bar{\Omega} + \sum_i F_i^T \Omega_i F_i, \quad (8)$$

where  $F_i$  is the projection matrix that implements operations that allocate  $\xi_i$  and  $\Omega_i$  at the correct entries for  $\xi$  and  $\Omega$ .

Finally, the robot checks if the updated  $\Omega$  satisfies the sparsification bound  $\theta_x$ . If not, the active beacons with the strongest links are selected and the weakest links are deactivated. Note that measurements from both active and non-active beacons are integrated in the update stage. Integrating measurements from a non-active beacon gives this beacon the possibility of being selected in the sparsification step.

### 3.2 Operation of beacons

The operation of the beacons is summarized in Alg. 2. Each beacon  $b_i$  that received an  $UpdateReq$  message first measures the range to the beacons in its sensing range –its measurement set is  $MS_i$ . The operation of  $b_i$  is different if it is at the “initialization phase” or at the “state vector phase”. If  $b_i$  is at the “state vector phase”, it computes its SLAM update contribution integrating the measurements in  $MS_i$  as follows:

$$\xi_i = \sum_{k \in MS_i} H_{i,k}^T R^{-1} [z_{i,k} - h_{i,k}(ev_i) + H_{i,k} ev_i] \quad (9)$$

$$\Omega_i = \sum_{k \in MS_i} H_{i,k}^T R^{-1} H_{i,k} \quad (10)$$

where  $h_{i,k}(ev_i)$  and  $H_{i,k}$  are the predictions and Jacobians for each measurement in  $MS_i$  either if it is a robot-beacon measurement or an inter-beacon measurement.  $R$  is the covariance matrix of the measurement noise.

Next, each beacon  $b_i$  transmits the resulting  $\xi_i$  and  $\Omega_i$  to the robot in an  $UpdateResp$  message, as in step 6 in Alg. 2.

If  $b_i$  is the “initialization phase”, the measurements in  $MS_i$  cannot be used for the update of the SLAM state vector: they are used for the initialization of beacon  $b_i$ . Two versions of the proposed scheme were developed:

**Algorithm 2** Summary of the operation of beacon  $b_i$ 


---

```

1: Receive UpdateReq message. Extract LNB
2: if ( $b_i$  is present in LNB) then
3:   Take measurements  $MS_i$ 
4:   if ( $b_i$  is at “state vector phase”) then
5:     Compute  $\xi_i$  and  $\Omega_i$  with  $MS_i$  as in (9) and (10)
6:     Send  $\xi_i$  and  $\Omega_i$  to robot in an UpdateResp message
7:   else
8:     Send  $MS_i$  to robot in an UpdateResp message
9:   end if
10: else
11:   Send to robot a BeaconDiscovery message
12: end if

```

---

- *Method1*: The initialization of each beacon  $b_i$  is computed by the robot. In this case  $b_i$  transmits to the robot an *UpdateResp* message containing  $MS_i$ . The robot will use  $MS_i$  for the initialization of  $b_i$ . When beacon initialization is finished, the robot computes the estimation of the location of  $b_i$  and adds it to the SLAM state vector  $\mathbf{x}$ .
- *Method2*: Each beacon computes its own initialization: both SLAM update and beacon initialization are decentralized.  $b_i$  integrates  $MS_i$  in its own initialization tool. When the beacon initialization is finished,  $b_i$  sends the robot its estimation in an *UpdateResp* message so that it adds the initialized beacon to  $\mathbf{x}$ .

The selection between *Method1* or *Method2* depends on the beacon computational capabilities and on the initialization tool. For instance, with beacons implemented using Wireless Sensor Networks (WSN) technology, *Method2* is suitable in case of using trilateration initialization but would require *Method1* if using PFs. Beacons implemented with embedded PCs (*RaspberryPi* as in the experiments), smartphones or PDAs can execute *Method2* using PFs as initialization tools.

#### 4 Analysis of the proposed scheme

If we analyze the building of the information matrix in the proposed scheme we can obtain the following conclusions:

- In the proposed scheme the integration of inter-beacon measurements is responsible for obtaining higher values of  $\Omega$ , and hence lower robot localization and map uncertainties, than schemes that integrate only robot-beacon measurements. We believe that this conclusion is clear and for brevity it is not proven.
- The proposed method preserves the sparsity of the information matrix of the conventional SEIF SLAM. Under common assumptions, both create the same links of the information matrix. This will be proven in Section 4.2.
- The proposed scheme preserves the constant time property of SEIF SLAM, which integrates only robot-beacon measurements. It is clear that the prediction stage and the distributed update stage of the proposed scheme are

constant time. In Section 4.3 it will be proven that state recovery in our scheme is also constant time.

Finally, in Section 4.4 the assumptions of the demonstrations used in Sections 4.2 and 4.3 are relaxed and the above conclusions are confirmed using simulations. Next, the building of the information matrix and links is described.

#### 4.1 Building of the information matrix

The information matrix and its structure was briefly presented in Section 2.1. Each off-diagonal entry of the information matrix constraints two elements in the state vector –each on-diagonal entry constraints one. The off-diagonal entry  $\Omega_{r,i}$  links together the estimations of the robot pose and of the location of beacon  $b_i$ . Entry  $\Omega_{i,j} \forall i \neq j$  links together the localization estimations of  $b_i$  and  $b_j$ .

Two types of links can be distinguished: robot-beacon links and inter-beacon links. A robot-beacon link relates the robot with a beacon. They are created or reinforced (the value of the link is incremented) when a robot-beacon measurement is integrated in the SLAM update stage. Integrating  $z_{r,j}$  affects  $\Omega_{r,r}$ ,  $\Omega_{j,j}$  and the robot-beacon links between the robot and  $b_j$ , i.e.  $\Omega_{r,j}$  and its symmetric  $\Omega_{j,r}$ . Inter-beacon links relate two beacons. They are created or reinforced when an inter-beacon measurement is integrated. Integrating  $z_{i,j}$  affects  $\Omega_{i,i}$ ,  $\Omega_{j,j}$  and inter-beacon links  $\Omega_{i,j}$  and  $\Omega_{j,i}$ . Inter-beacon links are created/reinforced also when integrating the robot motion in the SLAM prediction stage.

Figure 3 illustrates how the information matrix is built in the conventional SEIF SLAM (a-c) and in the proposed scheme (d-f) in a simple example. We assume that  $b_1$  and  $b_2$  are currently direct beacons (within  $SR_r$ ) and that  $b_1$  and  $b_2$  are within the sensing region one another.  $b_3$  is within the sensing region of  $b_2$ . We assume that initially  $\Omega_{r,1}=\Omega_{r,2}=\Omega_{1,2}=0$ . First, we describe the operation of the conventional SEIF SLAM. The SEIF SLAM update stage integrates measurements  $z_{r,1}$  and  $z_{r,2}$ , which creates robot-beacon links  $\Omega_{r,1}$  and  $\Omega_{r,2}$  –and their symmetric links, see Fig. 3-a. Next, the robot motion in the SEIF SLAM prediction stage transfers some low amount of information from the robot-beacon links  $-\Omega_{r,1}$  and  $\Omega_{r,2}$  in Fig. 3-b– to the inter-beacon links between the involved beacons  $-\Omega_{1,2}$ . Link  $\Omega_{1,2}$  is created. At that time,  $\Omega_{r,1}$  and  $\Omega_{r,2}$  are strong (have high value) whereas  $\Omega_{1,2}$  is weak. Next, the sparsification step in this example deactivates  $b_1$  and hence, removes link  $\Omega_{r,1}$ , see Fig. 3-c.

In the proposed distributed scheme the update stage integrates  $z_{r,1}$ ,  $z_{r,2}$ ,  $z_{1,2}$  and  $z_{2,3}$ , which strengthens robot-beacon links between the robot and all direct beacons  $-\Omega_{r,1}$ ,  $\Omega_{r,2}$ – and also the involved inter-beacon links  $-\Omega_{1,2}$  and  $\Omega_{2,3}$ , see Fig. 3-d. The prediction stage transfers some low amount of information from  $\Omega_{r,1}$  and  $\Omega_{r,2}$  to  $\Omega_{1,2}$ , see Fig. 3-e. At that

time, links  $\Omega_{r,1}$ ,  $\Omega_{r,2}$ ,  $\Omega_{1,2}$  and  $\Omega_{2,3}$  are strong. Finally, the sparsification step deactivates  $b_1$  and removes  $\Omega_{r,1}$ , Fig. 3-f.

Our scheme creates exactly the same robot-beacon links that SEIF SLAM and exactly with the same value –strength. The main difference between both is how they treat inter-beacon links. In SEIF SLAM the only way to create/reinforce inter-beacon links is through the robot motion in the prediction stage. Besides, in our method they are created/reinforced also when integrating inter-beacon measurements. As a result inter-beacon links accumulate significantly more information than in SEIF SLAM.

#### 4.2 Preservation of the sparsity of the information matrix

The conventional SEIF SLAM imposes sparsification bounds both in the number of robot-beacon links,  $\theta_x$ , and in the number of inter-beacon links,  $\theta_y$  [25]. In [25] it was proven that keeping the number of robot-beacon links below  $\theta_x$  automatically constrains the number of inter-beacon links below  $\theta_y$ . One could think that in our scheme integrating inter-beacon measurements may increase the number of inter-beacon links, violating the sparsification bound. In this section we prove that this is not the case: we demonstrate that in many scenarios it creates exactly the same inter-beacon links that the conventional SEIF SLAM. Our scheme accumulates higher amount of information than the conventional SEIF SLAM, but this increment is not used to create new links but to increase the strength of existing links, enabling lower map and robot uncertainty with a compact description.

Without loss of generality we assume that: (1) every beacon is within the robot sensing range at some time in the robot path and; (2) every pair of neighbor beacons in the environment (beacons which are within the sensing region one another) fall simultaneously within  $SR_r$  at some time in its path. These assumptions do not involve practical con-

straints. They are met in cases where the robot follows dense-navigation paths, which are typically adopted in SLAM in mapping or surveillance applications. Besides, in Section 4.4 it is shown that sparsity preservation is confirmed also when these assumptions are relaxed.

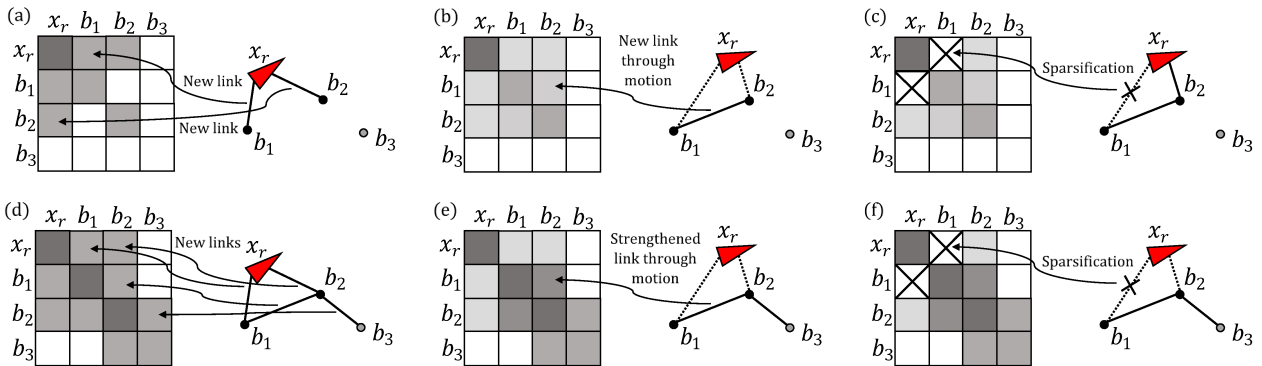
In the proposed scheme inter-beacon links are created through the robot motion and also by the integration of inter-beacon measurements. In the following the sets of inter-beacon links created in the full experiment through both mechanisms are analyzed. The integration of  $z_{i,j}$  creates/ reinforces inter-beacon link  $\Omega_{i,j}$ . Assuming dense-navigation robot paths, the robot will integrate inter-beacon measurements of every pair of neighbor beacons  $\forall b_i, b_j : b_i \in SR_j$ . Thus, in the full robot path the set of inter-beacon links created by inter-beacon measurements are:

$$L_{IB} = \{\Omega_{i,j} > 0 \quad \forall b_i, b_j : b_i \in SR_j\} \quad (11)$$

Besides, recall that the robot motion creates an inter-beacon link between any pair of beacons in which both beacons fall simultaneously within  $SR_r$ . The robot path is assumed dense enough such that every pair of neighbor beacons  $b_i$  and  $b_j$  in the environment  $\forall b_i, b_j : b_i \in SR_j$  fall simultaneously within  $SR_r$  at some time in the robot path. Thus, the set of inter-beacon links between neighbor beacons created through the robot motion in the full robot path is:

$$L_R^1 = \{\Omega_{i,j} > 0 \quad \forall b_i, b_j : b_i \in SR_j\} \quad (12)$$

The robot motion creates inter-beacon links between neighbor beacons but, also between beacons that are not neighbors one another but both are simultaneously within  $SR_r$  at a certain time  $t$ . The set of inter-beacon links between non-neighbor beacons will be called  $L_R^2$ . In the general case  $L_R^2 \neq \emptyset$  and the set of inter-beacon links created through the



**Fig. 3** Top) Effect of the steps in the conventional SEIF SLAM on the information matrix: (a) measurement integration, (b) prediction stage and (c) sparsification. Bottom) Effect of the steps in the proposed scheme on the information matrix : (d) measurement integration, (e) prediction stage and (f) sparsification.

robot motion is  $L_R = L_R^1 \cup L_R^2$ . From Eqs. (11) and (12) it is easy to notice that  $L_{IB} \subseteq L_R$ .

In the conventional SEIF SLAM inter-beacon links are created only via the robot motion: the set of inter-beacon links is  $L_{SEIF} = L_R$ . In our scheme they are created through both mechanisms: the set of inter-beacon links is  $L_{prop} = L_{IB} \cup L_R = L_R$ . Thus, our scheme creates exactly the same inter-beacon links as SEIF SLAM, preserving the sparsity of the information matrix. From Section 4.1 it is clear that both create the same robot-beacon links. Thus, it is concluded that under the above assumptions both create the same robot-beacon and also inter-beacon links.

If the dense-navigation robot path assumption is not met, inter-beacon measurements could create some links in  $L_{IB}$  that are not present in  $L_R$ . Section 4.4 evaluates through simulation the effects of relaxing this assumption.

#### 4.3 Preservation of constant time property

Constant time execution is a well known property of SEIF SLAM. The proposed scheme distributes measurement gathering and update computation making them independent of the map size. This section analyzes if the state/map recovery in the proposed scheme preserves constant time execution.

A critical key for SEIF SLAM being constant time is that it only recovers the state of the robot and of all the active direct beacons. As described in Section 3.1 our scheme recovers  $ev_i$  for each direct beacon  $b_i$  currently within  $SR_r$ . Recalling Eq. (6), state/map recovery for direct beacon  $b_i$  involves computing:  $\mu_r$ ,  $\mu_i$  and the state of the beacons neighbors of  $b_i$ . This should be done for each direct beacon at time  $t$ . Thus, considering all direct beacons it is easy to notice that at time  $t$  state/map recovery should recover: the robot state; the states of currently active direct beacons and; the state of active indirect beacons at time  $t$ —recall the definition in Section 3. In the proposed scheme recovering the states of indirect beacons involves an increment in the computational cost over that of SEIF SLAM. However, as proven below state/map recovery is kept constant time.

Let  $n_{DB}$  and  $n_{IB}$  be the number of direct and indirect beacons in the surrounding of the robot at time  $t$ . SEIF SLAM recovers the states of the robot and of direct beacons (in black in Fig. 2). The proposed scheme recovers also the state of indirect beacons (in gray in Fig. 2). Our scheme uses the state recovery algorithm presented in [23], which complexity is linear with the number of beacons states recovered:  $O(n_{DB})$  for SEIF SLAM and  $O(n_{DB} + n_{IB})$  for the proposed scheme.

The complexity of state/map recovery for the proposed method is as bounded as it is for SEIF SLAM. The computational burden of state/map recovery in our method—and also in SEIF SLAM—depends on beacon density, and not on the total number of beacons present in the environment. Highly

inhomogeneous local beacon densities are not very useful for RO-SLAM. It is more interesting if beacons are deployed in densities with some homogeneity. In these cases  $n_{DB} + n_{IB}$  will have similar values along the robot path.

#### 4.4 Simulation relaxing assumptions

This section analyzes using simulations the conclusions in Sections 4.2 and 4.3 when the assumptions are relaxed. A  $70 \times 70m$  scenario with 50 randomly deployed beacons was considered. The range measurement noise is assumed Gaussian with zero mean and standard deviation of  $\sigma_m = 0.8m$ . The robot trajectories were different in every experiment and did not meet the dense-navigation assumption. The sparsification bound was  $\theta_x = 10$ , in the range suggested in [23]. The proposed method was compared with EKF SLAM and SEIF SLAM. All the methods used PFs with 500 particles for beacon initialization and of course, all were set with exactly the same parameters.

It is easy to notice that the proposed scheme integrates more information than EKF SLAM and SEIF SLAM. Table 1 compares their performance in a set of 50 simulations with different beacon settings and robot paths. Two main metrics are analyzed: the average amount of information integrated in  $\Omega$  and the average number of inter-beacon links created. Our scheme creates the same robot-beacon links than SEIF SLAM and are not compared. In average, the proposed method integrated 66% more information than EKF SLAM and 67% more than the conventional SEIF SLAM. Besides, in average EKF SLAM created 1042 inter-beacon links, SEIF SLAM created 631 and the proposed scheme, 645. These scenarios do not meet the dense-navigation robot path assumption and the proposed scheme created only 2.2% more inter-beacon links but integrated 67% more information than the conventional SEIF SLAM.

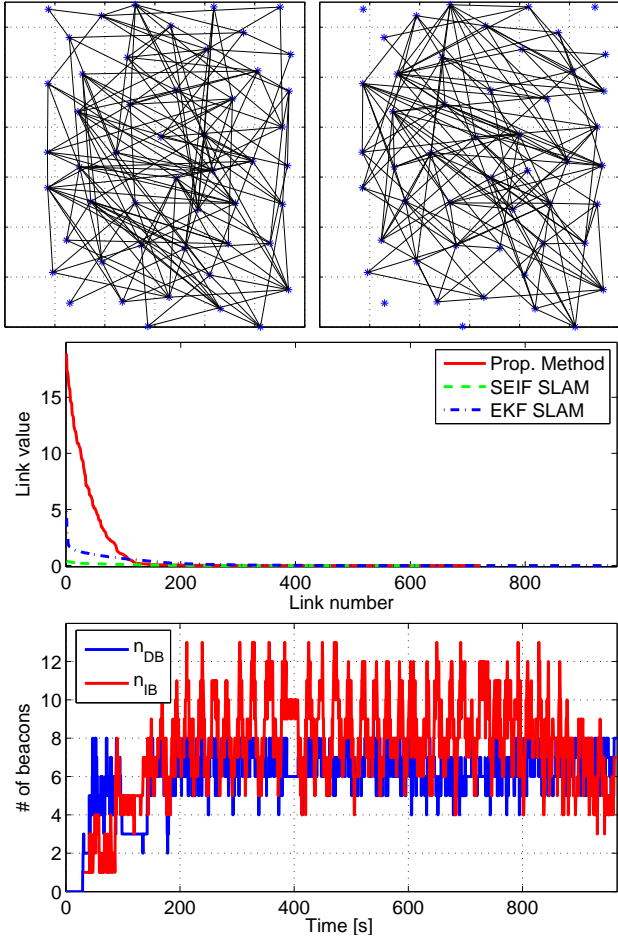
**Table 1** Comparison of the average amount of information accumulated and the number of inter-beacon links created by EKF SLAM, SEIF SLAM and the proposed scheme in a set of experiments.

	EKF SLAM	SEIF SLAM	<i>Prop. scheme</i>
$-\log( \Omega )$	197.37	197.03	328.41
inter-beacon links	1042	631	645

The proposed method creates the same inter-beacon links as the conventional SEIF SLAM in case of having dense-navigation robot paths. If not, it creates links involving beacons that are out of the sensing range of the robot in its path. These links could not be created by SEIF SLAM and are useful to improve map estimation completeness. For instance, Fig. 4-top shows the inter-beacon links created by SEIF SLAM and the proposed method in one example. In the proposed scheme all the beacons created links with other



beacons. That is not the case with SEIF SLAM, which cannot create links with some beacons originating incomplete maps.



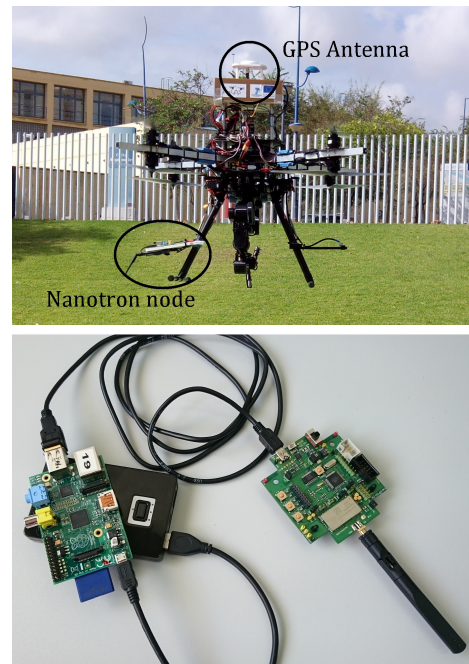
**Fig. 4** (Top) Inter-beacon links created in one simulation by the proposed method (left) and SEIF SLAM (right). Only the 25% strongest links are shown for visibility. (Center) Values of the inter-beacon links created by the proposed method, SEIF SLAM and EKF SLAM, ordered –stronger links first– for visualization. (Bottom) Values of  $n_{DB}$  and  $n_{IB}$  along the simulation.

Our method creates a relatively low number of strong inter-beacon links between nearby beacons which provide high amount of information of the map. As an example, Fig. 4-center shows the values of the inter-beacon links created in each case in the simulation in Fig. 4-top. The links resulting in each case are ordered –stronger links first– for visualization. Similar conclusions were obtained in the rest of scenarios. The complexity of state/map recovery in SEIF SLAM and the proposed scheme are respectively  $O(n_{DB})$  and  $O(n_{DB} + n_{IB})$ . The values of  $n_{DB}$  and  $n_{IB}$  along the example in Fig. 4-top are shown in Fig. 4-bottom. Similar conclusions were obtained in the rest of simulations. It can be noticed that  $n_{IB}$  is higher when  $n_{DB}$  is higher, and lower when  $n_{DB}$  is lower, exhibiting some proportionality.  $n_{DB}$  took values between 4 and 8 during most of the time, and  $n_{IB}$ , between

5 and 13. The lowest values took place when the robot was located at the borders of the scenario, where few beacons were within  $SR_r$ . The highest occurred the robot was at places with high local beacon densities. These results depend on beacon density and not on map size.

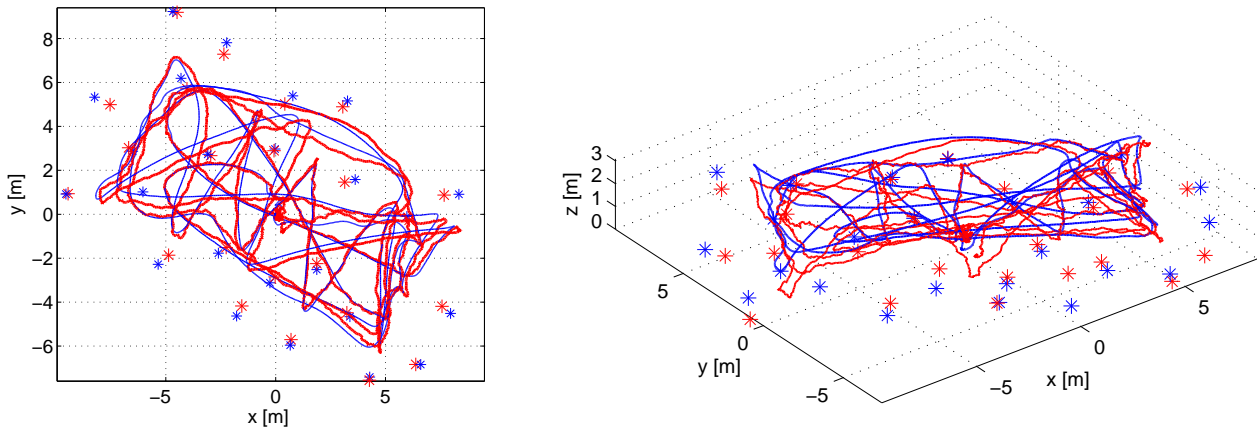
## 5 Validation in 3D SLAM experiments

The proposed scheme was integrated and experimented in AMUSE, an octorotor aerial vehicle developed in the ARCAS project for aerial manipulation and industry maintenance [14], see Fig. 5-top. Maintenance of industrial facilities is currently performed using sensor nodes that gather measurements for process monitoring and anomaly detection. In these complex scenarios GPS is often unavailable or has bad quality –intermittent signals, reception from few satellites, GPS shadows. UAS are suitable tools for confirming the detected anomalies but require accurate sensor and robot localization.



**Fig. 5** (Top) AMUSE octorotor flying during one experiment. (Bottom) Components of the beacons employed: a *RaspberryPi* connected to a *Nanotron* sensor, a WiFi USB adapter and a battery.

A total of 24 beacons were deployed at random locations and different heights in a 25x25 m environment. Each beacon was comprised of a *RaspberryPi* running Linux connected through USB to a *Nanotron nanoPAN 5375* Time-of-Flight (ToF) range sensor and to a WiFi USB adapter, see Figure 5-bottom. The performance of *Nanotron* sensors in outdoors was characterized experimentally. Their error can be modeled as a Gaussian PDF with zero mean and a standard deviation of  $\sigma_m = 0.6m$ . Each beacon executed an independent ROS



**Fig. 6** Result of 3D SLAM experiment: (left) XY view, (right) 3D view. The mean map and robot errors were 0.34 and 0.49 m, respectively. The mean errors in XY were 0.19 and 0.28 m.

(Robots Operating System) node that implemented the operation shown in Alg. 2. The PFs were set with 500 particles distributed on a sphere. The sparsification bound taken was  $\theta_x = 10$ , in the range suggested in [23]. A *Nanotron nanoPAN 5375* sensor was mounted on the octorotor landing skid. AMUSE is equipped with a *Novatel OEM6* GPS with centimeter accuracy, which was taken as ground truth. UAS odometry obtained from Inertial Measurement Units is often too noisy to be used in SLAM. Like most works in 3D SLAM, e.g. [10], we opted for employing 5 beacons as anchors for correcting UAS odometry.

### 5.1 Performance analysis

Figure 6 shows the result of the proposed scheme in one experiment in XY view (left) and 3D view (right). The resulting average map and robot errors were 0.34 and 0.49 m, respectively. The errors in XY were 0.19 and 0.35 m. In Z, they were 0.29 and 0.28 m. Table 2 compares the proposed *Method2* with EKF SLAM and SEIF SLAM. Of course, all the methods were set with exactly the same parameters. This table shows the resulting average map and robot localization RMS errors and PF convergence times. The mean robot and map RMS errors for *Method2* were 21% and 33% lower than for SEIF SLAM, and 20% and 30% lower than for EKF SLAM. The mean PF convergence time for *Method2* was 81% lower than for SEIF SLAM. Besides, the robot CPU time for *Method2* was 35.3% lower than for SEIF SLAM. EKF SLAM devoted the lowest robot CPU time: it is efficient with low number of beacons but scales badly with the map size as described in Section 5.2.

If we compare these results in 3D experiments with those in 2D described in [26], we notice that in the 3D experiments the robot location error was 49 cm –35 cm in XY– and, in the 2D experiments, it was 41 cm. There are two main differences between both sets of experiments. First, the range

**Table 2** Performance comparison of EKF SLAM, SEIF SLAM and the proposed *Method2*.

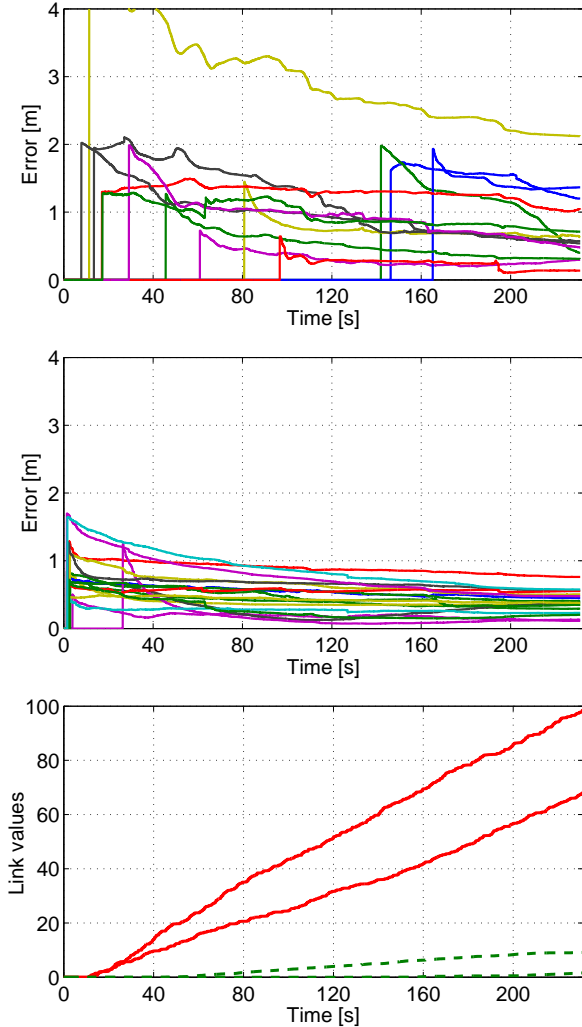
	EKF SLAM	SEIF SLAM	Prop. <i>Method2</i>
Map RMS error [m]	0.49	0.51	0.34
Robot RMS error [m]	0.59	0.62	0.49
PF conv. time [s]	46.2	46.5	8.5

sensors were the same in both cases but the 3D experiments were performed in outdoors and the 2D experiments, in indoors. The range measurements from Nanotron sensors have higher noise level in indoors ( $\sigma_m = 0.8m$ ) than in outdoors ( $\sigma_m = 0.6m$ ). Second, the odometry of the aerial robot is based essentially on measurements from Inertial Measurement Units (IMU) and is significantly more noisy than that of the ground robot, which is based on encoders. The robot odometry is used in the prediction stage and range measurements, in the update stage. Thus, the prediction stage in these 3D experiments is less accurate than that in [26] but the update stage is more accurate. If we consider both effects, in these sets of experiments the improvement due to lower measurement noise was stronger than the worsening due to poorer odometry.

Figures 7-top and 7-center show the evolution of the location error for each beacon in *Method2* and SEIF SLAM –EKF SLAM performs similarly to SEIF SLAM and is not shown. The drawing for each beacon starts when its auxiliary PF converged. In the proposed *Method2* beacon PFs converge significantly sooner and with lower error than in SEIF SLAM.

In these experiments, the proposed method created exactly the same inter-beacon links –143– as SEIF SLAM. The sensing range of *Nanotron* sensors was higher than 100 m and hence, the robot dense-navigation assumption was met. Figure 7-bottom analyzes the evolution of the values of each inter-beacon link along the experiment in the proposed scheme (red) and in SEIF SLAM (green). For better illustrating the tendency, the envelope that groups the 90%

central curves is shown for each case. Links start taking non-zero values in the proposed scheme sooner than in SEIF SLAM, which is originated by the shorter beacon PF convergence times. The values of inter-beacon links in the proposed scheme grow at higher constant rate –and are higher at any time– than in SEIF SLAM. At any time along the experiment the proposed method has accumulated more information in  $\Omega$ , involving lower mapping uncertainty.

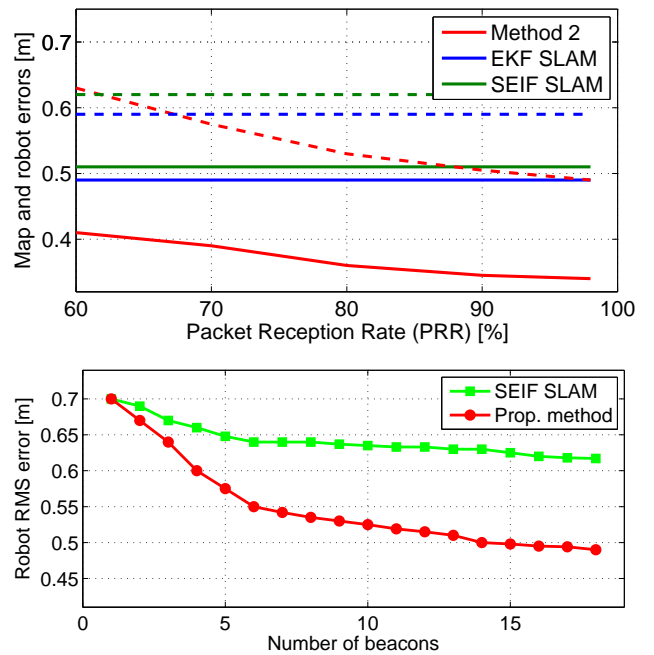


**Fig. 7** Evolution of the location error for each beacon in SEIF SLAM (top) and in the proposed *Method2* (center). (Bottom) Evolution of the values of inter-beacon links along the experiment in SEIF SLAM (green) and in our scheme (red). For better illustrating the tendency, the envelope that groups the 90% central curves is shown.

## 5.2 Discussion

This section analyzes four aspects of the proposed scheme: (1) robustness against message errors, (2) beacon density, (3) consumption of resources of beacons and (4) scalability.

The impact of message loss was evaluated using the measurements collected in the real experiments and simulating transmission errors with Packet Reception Rates (PRRs) in the range [60%-99%]. Figure 8-top shows the map and robot localization errors obtained with EKF SLAM, SEIF SLAM and the proposed *Method2*. The first two are not influenced by transmission errors. The update stage of *Method2* is additive, which makes it rather robust to message loss. Map RMS error for *Method2* is lower than for EKF SLAM and SEIF SLAM for any PRR level. *Method2* had the lowest robot error for PRR levels higher than 70%. Although the proposed method makes extensive use of transmissions, it is significantly robust to transmission errors.



**Fig. 8** (Top) Impact of PRR on map –full lines– and robot –dashed– errors for the proposed *Method2*, EKF SLAM and SEIF SLAM. (Bottom) Analysis of the influence of beacon density on the proposed *Method2*.

Beacon density has also influence in the proposed method. Rather than physical density, in our problem density refers to measurement density, i.e. in average the number of beacons that are within the measurement range of other beacons. The measurement range of the beacons used in these experiments is 100 m, i.e. each sensor could measure the range to the rest of the deployed sensors. Hence, reducing the number of beacons considered in the experimental area reduces the beacon density. In the following the influence of beacon density in real experiments is analyzed. We executed SEIF and the presented method with the logs of the experiments shown in Section 5.1 but with lower numbers of sensors  $ND = 1, 2, \dots, 18$ , i.e. we did not considered the measurements from some of the nodes. For each  $ND$ , both

techniques were executed in a total of 15 different scenarios, each with different combinations of beacons (randomly selected). Figure 8-bottom shows the robot RMS location errors for each  $ND$ . With low  $ND$ , i.e. low beacon densities, the improvement of the robot RMS error with density is clear: changing from  $ND=2$  to  $ND=6$  originates informative inter-beacon measurements that significantly reduce error. With higher beacon densities, the improvement with density is not very evident: the errors decrease at lower rate since the main component in the error is originated by the robot odometry. SEIF behaves similarly but its improvement is lower due to the lack of inter-beacon measurements.

In these experiments beacon density of  $ND=6$  was a good trade-off between robot location error and beacon deployment effort. In a real environment, to be conservative we would select  $ND=8-11$ . Since the measurement range of the sensors is 100 m, it involves deploying 8-11 beacons per  $100 \times 100 \text{ m}^2$ , which is low effort-demanding and feasible in many industrial plants. In the experiments performed the sensor locations were not carefully selected: they were selected in a random way. In a real scenario no special care would be required either apart from trying to avoid large direct obstacles between beacons and the robot.

The proposed scheme exploits beacon capabilities involving higher consumption of beacon resources than conventional methods. Table 3 shows the average energy consumed by the beacons in the experiments. It was estimated with the number of measurements gathered and the number of messages interchanged in the experiments, using the power consumption characteristics in the manufacturers data-sheet. The beacons energy consumption using *Method2* is more than 3 times higher than with EKF SLAM and SEIF SLAM. It seems interesting to develop strategies to reduce the number of inter-beacon measurements without losing their advantages: map and robot accuracy and fast map initialization.

**Table 3** Average beacon energy consumption [J].

EKF SLAM	SEIF SLAM	Proposed <i>Method2</i>
182.55	182.26	634.15

In order to analyze scalability we performed series of simulations, with the same parameters of the real experiments. 25 experiments were performed for each map size with between 20 and 200 beacons –the beacon density was kept steady. Figure 9-a shows the average robot computational burden of the EKF SLAM, SEIF SLAM and the proposed *Method1* and *Method2*. *Method2* is the most efficient for maps with more than 50 beacons. For any map size *Method2* consumes lower robot CPU time than *Method1* and, *Method1*, lower than SEIF SLAM. Figure 9-b shows the map and robot errors obtained. The proposed *Method2* obtains the lowest errors. *Method1* and *Method2* obtain similar errors: only *Method2*

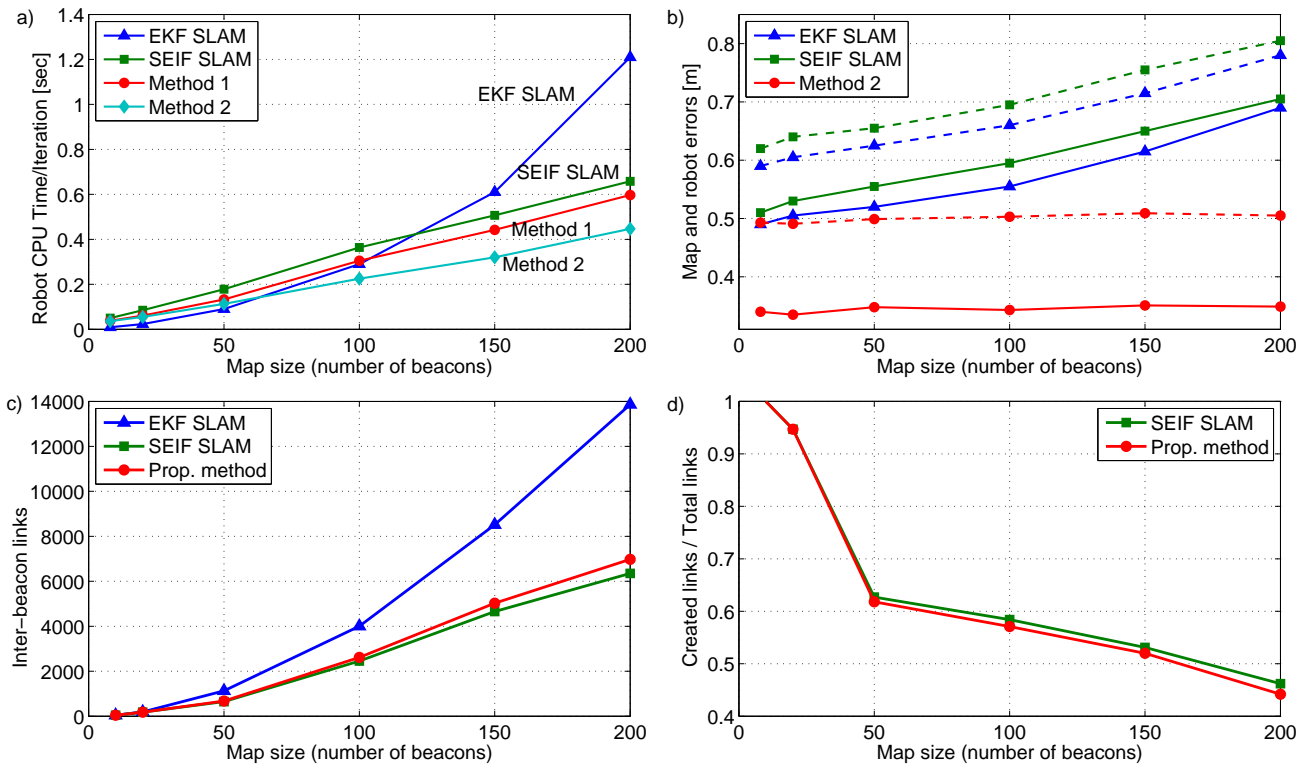
is shown for clarity. The advantage is more evident with larger maps. Inter-beacon measurements enforce map consistency and, in absence of inter-beacon measurements, the estimations are more influenced by the robot odometry noise.

Figure 9-c shows the average number of inter-beacon links created at the end of each simulation by each method. Figure 9-d shows the sparsity of the information matrix expressed as the ratio between the inter-beacon links created in the experiment and the total number of possible inter-beacon links. The sparsity of the information matrix increases with the map size. They both perform similarly in terms of the absolute number of links and sparsity of the information matrix. Notice that although with higher map sizes the number of links created in the proposed scheme is a bit higher than in SEIF SLAM, its information matrix is sparser. Integrating inter-beacon measurements allows our scheme to detect more beacons and add them to the state vector, improving map completeness, but these new beacons create very few links, increasing the sparsity of the information matrix. For instance, with maps of 190 beacons, the state vector in our scheme contained 185 beacons, created 7048 inter-beacon links out of the total of 17020 possible links, whereas in SEIF SLAM the state vector contained 167 beacons and created 6349 inter-beacon links out of 13861 possible links. Our scheme builds more complete maps involving larger state vectors and information matrices than SEIF SLAM but also the information matrices are sparser and the robot CPU burden is lower as shown in Figure 9-a.

## 6 Conclusions

This paper proposed a distributed SEIF SLAM scheme for robot-sensor network cooperation applications. This cooperation allows the distribution between the robot and the sensor network nodes (beacons) of tasks like measurement gathering and integration in SLAM. Beacons take range measurements to the robot and to their neighbor beacons, estimate their contribution to the SLAM update and send it to the robot. Besides robot-beacon measurements, the proposed scheme naturally integrates inter-beacon measurements, resulting in more accurate map and robot estimations and faster convergence of the beacon initialization PFs. Distributing measurement integration in SLAM reduces the robot burden.

The impact of the proposed scheme on the building of the information matrix was analyzed in detail. It was proven that: (1) it accumulates higher amount of information than SEIF SLAM; (2) it creates a very similar number of links than SEIF SLAM, preserving the sparsity of the information matrix and; (3) it preserves the constant time property of SEIF SLAM. The proposed scheme integrates more information than SEIF SLAM preserving its compact description, efficiency and scalability. The proposed scheme was validated and compared to existing methods in 3D real experiments.



**Fig. 9** Performance analysis of the proposed *Method1*, *Method2*, EKF SLAM and conventional SEIF SLAM with different map sizes: (a) robot computational burden measured using MATLAB profiler; (b) map –represented with full lines– and robot –dashed lines– errors; (c) number of inter-beacon links created by each method; (d) sparsity of the information matrix in terms of the ratio between created links and total possible links.

Integrating inter-beacon measurements involves significant advantages but also requires the consumption of resources such as energy and bandwidth. It would be interesting to develop tools that decide which measurements are more interesting to integrate in SLAM and which do not provide enough information and are not worth gathering them. This issue is object of current research.

**Acknowledgements** The authors thank A. Jiménez-Cano, V. Vega and J. Braga for their help and support in the 3D experiments.

## References

- Banatre, M., Marron, P., Ollero, A., Wolisz, A.: Cooperating Embedded Systems and Wireless Sensor Networks. Wiley (2008)
- Blanco, J.L., Fernandez-Madrigo, J.A., Gonzalez, J.: Efficient probabilistic range-only slam. In: Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS, pp. 1017–1022 (2008)
- Brunner, C.J., Peynot, T., Vidal-Calleja, T.A., Underwood, J.P.: Selective combination of visual and thermal imaging for resilient localization in adverse conditions: Day and night, smoke and fire. *J. Field Robotics* **30**(4), 641–666 (2013)
- Challa, S., Leipold, F., Deshpande, S., Liu, M.: Simultaneous localization and mapping in wireless sensor networks. In: Intelligent Sensors, Sensor Networks and Information Processing Conf., 2005, pp. 81–87 (2005)
- Corke, P., Hrabar, S., Peterson, R., Rus, D., Saripalli, S., Sukhatme, G.: Deployment and connectivity repair of a sensor net with a flying robot. In: Experimental Robotics IX, *Springer Tracts in Advanced Robotics*, vol. 21, pp. 333–343. Springer (2006)
- Martinez-de Dios, J., Lferd, K., de San Bernabe, A., Nunez, G., Torres-Gonzalez, A., Ollero, A.: Cooperation between uas and wireless sensor networks for efficient data collection in large environments. *Journal of Intelligent & Robotic Systems* **70**(1-4), 491–508 (2013)
- Djugash, J., Singh, S., Grocholsky, B.: Decentralized mapping of robot-aided sensor networks. In: IEEE Intl. Conf. on Robotics and Automation, ICRA 2008, pp. 583–589
- Djugash, J., Singh, S., Kantor, G., Zhang, W.: Range-only slam for robots operating cooperatively with sensor networks. In: Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA, pp. 2078–2084 (2006)
- Eustice, R., Walter, M., Leonard, J.: Sparse extended information filters: Insights into sparsification. In: Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS, pp. 3281–3288 (2005)
- Fabresse, F., Caballero, F., Maza, I., Ollero, A.: Localization and mapping for aerial manipulation based on range-only measurements and visual markers. In: IEEE Intl. Conf. on Robotics and Automation, ICRA 2014, pp. 2100–2106
- Frese, U.: A proof for the approximate sparsity of slam information matrices. In: Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA, pp. 329–335 (2005)
- Frese, U.: Treemap: An  $o(\log n)$  algorithm for simultaneous localization and mapping. In: Spatial Cognition IV. Reasoning, Action, Interaction, pp. 455–477. Springer (2005)
- Frey, B.J., MacKay, D.J.: A revolution: Belief propagation in graphs with cycles. *Advances in neural information processing systems* pp. 479–485 (1998)
- Heredia, G., Jiménez-Cano, A., Sánchez, M., Llorente, D., Vega, V., Braga, J., Acosta, J., Ollero, A.: Control of a multirotor outdoor

- aerial manipulator. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS 2014
15. Liu, Y., Thrun, S.: Results for outdoor-slam using sparse extended information filters. In: Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA, vol. 1, pp. 1227–1233 (2003)
  16. Maza, I., Caballero, F., Capitan, J., Martinez-de Dios, J.R., Ollero, A.: A distributed architecture for a robotic platform with aerial sensor transportation and self-deployment capabilities. *Journal of Field Robotics* **28**(3), 303–328 (2011)
  17. Menegatti, E., Danieletto, M., Mina, M., Pretto, A., Bardella, A., Zanconato, S., Zanuttigh, P., Zanella, A.: Autonomous discovery, localization and recognition of smart objects through wsn and image features. In: Proc. IEEE GLOBECOM 2010, pp. 1653–1657
  18. Menegatti, E., Zanella, A., Zilli, S., Zorzi, F., Pagello, E.: Range-only slam with a mobile robot and a wireless sensor networks. In: IEEE Intl. Conf. on Robotics and Automation, ICRA, pp. 8–14 (2009)
  19. Nogueira, M., Sousa, J., Pereira, F.: Cooperative autonomous underwater vehicle localization. In: IEEE OCEANS 2010, pp. 1–9
  20. Olson, E., Leonard, J., Teller, S.: Robust range-only beacon localization. In: Proc. IEEE/OES Autonomous Underwater Vehicles, pp. 66–75 (2004)
  21. Paskin, M.A.: Thin junction tree filters for simultaneous localization and mapping. In: G. Gottlob, T. Walsh (eds.) Proc. of the 18th Intl. Joint Conf. on Artificial Intelligence, IJCAI, pp. 1157–1164. Morgan Kaufmann Publishers, San Francisco, CA (2003)
  22. Salas, M., Hussain, W., Concha, A., Montano, L., Civera, J., Montiel, J.: Layout aware visual tracking and mapping. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), pp. 149–156 (2015)
  23. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics, third edn. The MIT Press, Cambridge (2005)
  24. Thrun, S., Liu, Y.: Multi-robot slam with sparse extended information filters. In: *Robotics Research*, pp. 254–266. Springer (2005)
  25. Thrun, S., Liu, Y., Koller, D., Ng, A.Y., Ghahramani, Z., Durrant-Whyte, H.: Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research* **23**(7-8), 693–716 (2004)
  26. Torres-González, A., Martínez-de Dios, J., Ollero, A.: Efficient robot-sensor network distributed self range-only slam. In: IEEE Intl. Conf. on Robotics and Automation, ICRA 2014, pp. 1319–1326
  27. Torres-González, A., Martínez-de Dios, J., Ollero, A.: Integrating internode measurements in sum of gaussians range only slam. In: ROBOT2013: First Iberian Robotics Conference, pp. 473–487. Springer International Publishing (2014)
  28. Torres-González, A., Martínez-de Dios, J.R., Ollero, A.: An adaptive scheme for robot localization and mapping with dynamically configurable inter-beacon range measurements. *Sensors* **14**(5), 7684 (2014)