2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Efficient Robot-Sensor Network Distributed SEIF Range-Only SLAM

A. Torres-González, J. R. Martinez-de Dios and A. Ollero

Abstract— **This paper 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). Most existing RO-SLAM techniques consider beacons as passive devices disregarding the sensing, computing and communication capabilities they are actually endowed with. This paper proposes a Range-Only scheme based on Sparse Extended Information Filters (SEIF) that efficiently exploits their capabilities. The robot computes the SLAM prediction stage and distributes the update stage among beacons within its sensing area. The proposed scheme naturally integrates robot-beacon and inter-beacon measurements, significantly improving map and also robot estimations. Our scheme inherits from SEIF its efficiency and scalability and further reduces robot computational burden by exploiting the beacons computing capability. As a result, it has lower error and lower computer requirements than traditional methods. This paper presents the scheme, evaluates and compares its performance in simulations and real experiments.**

I. INTRODUCTION

We are interested in Range Only (RO) SLAM, which integrates range measurements between the robot and a set of static landmarks. RO-SLAM can be very useful in a wide range of applications such as advanced monitoring or search and rescue, in which robots cooperate with sensor nodes endowed with sensing, computing and communication capabilities. This interest has originated the development of a growing number of RO-SLAM methods that use sensor networks as landmarks [1], [2], [3]. However, the great majority of them consider beacons as passive devices disregarding many of their capabilities.

This paper presents an online SLAM scheme based on Sparse Extended Information Filters (SEIF) that efficiently exploits beacon capabilities by distributing measurement gathering and SLAM update computation among the beacons surrounding the robot. The robot performs SEIF prediction stage and broadcasts the predicted state. Each beacon receiving the message takes range measurements to each of the beacons within its sensing area and to the robot, computes the SEIF update integrating these measurements and sends the result back to the robot. The robot reconstructs the full updated state by adding the contributions it receives.

The proposed scheme integrates in the SEIF SLAM all direct robot-beacon and also inter-beacon measurements that involve at least one beacon within the robot sensing area, naturally avoiding repeated measurements. Integrating inter-beacon measurements reduces the map estimation uncertainty and indirectly also improves the robot localization accuracy. Also, the method is based on SEIF benefiting from its efficiency and scalability. Moreover, it exploits the computing capability of beacons reducing the robot computational burden and enabling its implementation in robots with lower computer capabilities. This paper presents the scheme and evaluates and compares its performance with traditional methods based on EKF and SEIF in simulations and experiments performed in the *CONET Integrated Testbed* [4].

The paper is organized as follows. Section I-A summarizes the main existing RO-SLAM methods. Section II presents the proposed method and motivates it with potential applications. Section III briefly summarizes SEIF SLAM as an introduction to our method, which is detailed in Section IV. Its evaluation and comparison with other existing methods is in Section V. Section VI shows its performance in real experiments. Conclusions are in Section VII.

A. Related work

Various SLAM filters have been applied to RO-SLAM. EKF and FastSLAM are maybe the most widely used methods. EKF SLAM [5], [6] is optimal in presence of Gaussian noise and obtains good results in most implementations. FastSLAM [7] 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 robot pose and the map.

In order to improve efficiency and scalability, different strategies have been developed in the past years. Many of them use the canonical form of the Gaussian distribution based on the information state vector and the information matrix. It has been demonstrated that many of the off-diagonal elements of the information matrix are relatively close to zero [8], [9]. Thin Junction Tree Filters [10] use an special data structure –called junction tree– to represent the information matrix. Work [10] proposes approximations to keep these junction tree simple –thin– regardless of the map size. Treemap Filters [11] hierarchically divide the map into local regions. The Sparse Extended Information Filter (SEIF) by Thrun et al. [8], [12] improves efficiency and scalability making approximations that maintain the information matrix sparse. SEIF for feature-based SLAM has been researched in some works. In [13] a modification for ensuring the

This work was supported by the EC-SAFEMOBIL (European Commission ICT-2011-288082), CLEAR (DPI2011-28937-C02-01) and the Ministerio de Educación y Deportes FPU Program. J. R. Martinez-de Dios thank the partial support from PLANET (European Commission FP7-257649-ICT-2009-5). The authors are with Robotics, Vision and Control Research Group, University of Seville, 41092 Seville, Spain arturotorres@us.es, jdedios@us.es, aollero@us.es

consistency of the global map estimate was proposed. SEIFs were used in multi-robot systems in [14]. We could not find any paper that applied SEIF to RO-SLAM.

Range measurements involve partial observability, i.e. several measurements are necessary to disambiguate a location. In RO-SLAM partial observability is addressed by two main approaches: delayed and undelayed mapping initialization. Undelayed initialization directly uses the measurements in the SLAM filter from the very beginning using multi-hypothesis estimation tools such as Sum of Gaussians [15], [7]. Delayed initialization employs auxiliaty tools to compute the initial estimation of landmarks. Examples of these tools are Particle Filters [6], Probability Grids [5], [16] and trilateration methods [2]. Trilateration methods, although simple and efficient, are very sensitive to measurement noise. It was the first approach but was soon overtaken since the SLAM performance is very sensitive to the accuracy in beacon initialization. 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.

All the aforementioned RO-SLAM methods use only robot-beacon measurements. Integrating range measurements between beacons is useful to improve the estimations of the map, and also, indirectly, of the robot. However, despite its potential advantages, very few methods exploiting inter-beacon measurements have been proposed. The general idea of using inter-beacon was given in [5], where different ways for incorporating inter-beacon measurements were proposed, mainly by using virtual nodes and adopting off-line map improvement using multidimensional scaling. These off-line SLAM solutions are not suitable for many applications, which require an on-line map and robot locations.

In most proposed methods the RO-SLAM algorithm is computed centralized by the robot. To the best of our knowledge no methods where beacons actively participate in the computation of the SLAM filter have been reported.

II. General description

Our work is motivated by schemes of cooperation between robots and sensor networks. Consider a GPS-denied scenario where a large number N of sensor nodes have been deployed at unknown locations. For instance, they have been randomly thrown for on-line monitoring a disaster or accident in an industry or urban area, where preexisting infrastructure can be damaged. The basic role of each sensor node is to periodically take measurements, e.g. toxic gas concentration, filter and compute statistics of the measurements and transmit them to a Base Station. For these tasks the nodes must be endowed with sensing, communication and computing capabilities. In fact, this is the case of most COTS sensor nodes [17]. Assume that each node is equipped with a range sensor and can measure the distance to the robot or to other nodes within its sensing area. Again, this is not a constraint. For instance, most COTS nodes can measure received signal strength (RSS) from incoming messages and can measure the range to the emitting node [17]. From now on, in the paper, the terms beacons and nodes will be used synonymously.

The application is to on-line monitor the status of the event. Having the precise location of the robot and each node is necessary for accurate monitoring. Besides, it enables advanced robot-sensor network cooperation strategies of interest in these applications such as using the robot for sensor node deployment [18], replacement [19] or collecting data from the sensors [20], among others. GPS cannot be used. A RO-SLAM method using sensor nodes (beacons) as landmarks can be very useful to on-line estimate nodes and robot locations. In this problem RO-SLAM can have advantages w.r.t. visual SLAM since suitable lighting conditions are not always present. Besides, RO-SLAM using beacons as landmarks naturally solves the data association problem.

This paper presents a distributed SEIF SLAM scheme that efficiently exploits beacons sensing, computing and communications capabilities. In contrast to traditional methods, which use only robot-beacon measurements, the proposed scheme naturally integrates also inter-beacon measurements significantly improving map and robot localization estimations. It is based on SEIF and scales well with the number of beacons. In contrast, EKF SLAM scales with $O(N^2)$ [21]. Also, most existing SLAM methods are executed centralized in the robot whereas our method distributes the SEIF SLAM update stage among beacons, reducing the robot computational burden.

The operation of our method is as follows. The SEIF SLAM prediction stage is executed in the robot. The robot keeps LNB, the list of beacons that are currently within the robot sensing area, SA_r. Then, the robot broadcasts an $UpdateReq$ message that includes LNB and the predicted state. A beacon b_i receiving the message, extracts LNB. If b_i is present in LNB, it takes range measurements to the robot ($z_{i,r}$) and to each of the beacons within its sensing area SA_i. This measurement set is designed as $MS_i = \{z_{i,r}, z_{i,j} : \forall b_j \in SA_i\}$, where $z_{i,j}$ is the range measurement to beacon b_j. Then, with MS_i beacon b_i computes the SEIF SLAM update and sends back the result to the robot in an $UpdateResp$ message. Finally, since the SEIF SLAM update stage is additive, the robot reconstructs the updated state by adding the contributions it received. To prevent overlapping of response messages the beacons can respond one after the other using their order in LNB.

Figure 1 illustrates its operation. In a traditional approach, see Fig. 1-top, the robot takes measurements to the beacons within SA_r –in grey– $\{z_{r,1}, z_{r,5}, z_{r,6}\}$. In our scheme, see Fig. 1-bottom, each beacon b_i in $LNB = \{b_1, b_5, b_6\}$ takes MS_i, e.g. b_5 takes $MS_5 = \{z_{5,r}, z_{5,1}, z_{5,6}\}$ and integrates them in the SEIF update. This scheme is equivalent to performing traditional SLAM with a fleet of virtual robots, one at each beacon within the sensing area of the real robot.

The proposed scheme naturally integrates all direct robot-beacon and inter-beacon measurements that involve at least one beacon within SA_r. It also naturally avoids repeated measurements. It takes one measurement for all these distances except for inter-beacon distances when both beacons

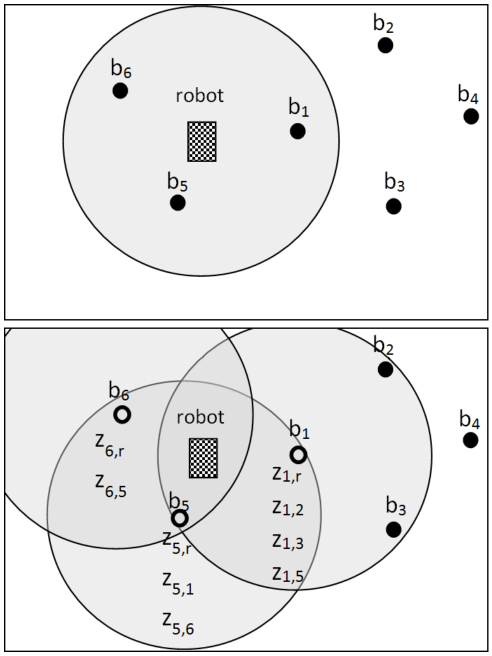

Fig. 1. Examples of a traditional RO-SLAM (top) and the proposed scheme (bottom). Grey circles represent sensing areas the robot (top) and of beacons b_1, b_5 and b_6 (bottom).

are within SA_r. In this case two different measurements are taken. As will be shown in Section V, the method is also robust to message loss since errors in messages of one beacon only affect the contribution of that beacon.

III. Range Only SEIF SLAM in a nutshell

This section briefly summarizes the SEIF SLAM algorithm as an introduction to the proposed method, which detailed in Section IV. The state vector $\vec{x}$ is comprised of the robot position and orientation (x_r) and the location of all the N beacons in the map ($x_1, x_2, \ldots, x_N$). The estimation of $\vec{x}$ is its mean μ. For notation simplicity, time subindex t has been omitted. For brevity, detailed descriptions of equations and algorithms have been obviated. They can be found in [21].

SEIFs are based on Extended Information Filters (EIFs). Duals to EKFs, EIFs represent the state vector by its information vector $\xi = \Sigma^{-1}\mu$ and the information matrix $\Omega = \Sigma^{-1}$. The geometry of Ω is in (1). Ω is symmetric, positive-semidefinite and each off-diagonal entry –called *link* [8]– represents the relation between two elements in $\vec{x}$.

$$\Omega = \begin{pmatrix} \Omega_{x_r,x_r} & \Omega_{x_r,x_1} & \cdots & \Omega_{x_r,x_N} \\ \Omega_{x_1,x_r} & \Omega_{x_1,x_1} & \cdots & \Omega_{x_1,x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_{x_N,x_r} & \Omega_{x_N,x_1} & \cdots & \Omega_{x_N,x_N} \end{pmatrix} \tag{1}$$

Integrating measurement $z_{r,j}$ in the EIF SLAM update stage only affects the on-diagonal elements Ω_{x_r,x_r} and Ω_{x_j,x_j} and the links between the robot and b_j, i.e. Ω_{x_r,x_j} and its symmetric Ω_{x_j,x_r}. Measurement updates increment the information of involved elements increasing the value of links in Ω –strengthen links. On the other hand, in the prediction stage noise in robot motion originates a loss of information in the relative location of the beacons, which decreases the value of the involved links –weakens links. Thus, at any time in the EIF SLAM operation some of the off-diagonal elements of Ω 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 EIF estimation than strong links but both involve similar computational burden. SEIF simplifies the structure of Ω by deleting weak links to save burden without significantly perturbing the EIF estimation.

SEIF operates as EIF but maintains a sparse representation of Ω by keeping the number of active beacons bounded by a threshold. At each step the discovered beacons are classified in active and passive. Active beacons are those that have active –non-zero– links. Every time the number of active beacons is above the bound, a sparsification step that deactivates the beacons with the weakest links is performed.

EIF measurement update modifies only the involved elements in Ω. Factorizing Ω allows efficient update stages regardless of the map size –numbers of beacons. SEIF inherits this efficiency. Moreover, by bounding the number of active beacons, SEIF significantly reduces the computational burden of the prediction stage. For linearizing the prediction and observation models it is required to recover the state estimate μ from the predicted $\bar{\Omega}$ and $\bar{\xi}$. Recovering the full state vector involves inverting a large matrix –with N beacons it is $O((2N+3)^3)$. Instead, SEIF recovers only the useful part of $\vec{x}$ –robot and active beacons– using an efficient algorithm [21]. Thus, SEIF significantly improves computational burden and scalability to the map size w.r.t. EIF. Enforcing sparseness in Ω involves an approximation error in the estimations obtained by SEIF. Work [21] suggests using sparsification bounds in the range $[4-10]$ to balance between accuracy degradation and burden reductions.

SEIF has been traditionally applied in visual and laser-based SLAM. The observation model adopted in our RO-SLAM SEIF for range measurement $z_{r,j}$ taken by the robot to beacon b_j is as follows:

$$h_{r,j}(\mu) = \sqrt{\delta_x^2 + \delta_y^2}, \tag{2}$$

where $\delta_x = \mu_x - \mu_x^j$ and $\delta_y = \mu_y - \mu_y^j$. The couples (μ_x, μ_y) and (μ_x^j, μ_y^j) are respectively the estimated positions of the robot and b_j. $h_{r,j}$ is nonlinear. Its Jacobian is:

$$H_{r,j} = \frac{\partial h_{r,j}}{\partial \mu} = \begin{bmatrix} \frac{\delta_x}{h_{r,j}} & \frac{\delta_y}{h_{r,j}} & 0 & \cdots & \frac{-\delta_x}{h_{r,j}} & \frac{-\delta_y}{h_{r,j}} & \cdots \end{bmatrix} \tag{3}$$

Range measurements have partial observability. The SEIF should be combined with a mechanism for beacon initialization. When the robot takes the first measurement from beacon b_j, the initialization of b_j is started and we say that beacon b_j is at the "Initialization stage". A number of measurements of b_j taken from different locations are required to compute an initial estimation of the location of the beacon. When the initial estimation is computed, it is incorporated in the state vector. At that moment we say that beacon b_j is at the "SEIF stage".

IV. Robot-sensor network distributed SEIF SLAM

The proposed method has two main differences from existing approaches: a) it distributes the SEIF computation among beacons; and b) it integrates robot-beacon and inter-beacon measurements. $h^{IB}_{i,j}$, the model used for inter-beacon measurement $z_{i,j}$, is the same as that used in (2) but taking $\delta_x = \mu^i_x - \mu^j_x$ and $\delta_y = \mu^i_y - \mu^j_y$. The Jacobian of $h^{IB}_{i,j}$ is:

$$H^{IB}_{i,j} = \begin{bmatrix} 0 & 0 & 0 & \cdots & \frac{\delta_x}{h_{i,j}} & \frac{\delta_y}{h_{i,j}} & \cdots & \frac{-\delta_x}{h_{i,j}} & \frac{-\delta_y}{h_{i,j}} & \cdots \end{bmatrix} \quad (4)$$

All the terms in $H^{IB}_{i,j}$ are zero except for the entries of the beacons involved in the measurement.

The operations of the robot and beacons in one prediction-update cycle are summarized in Algs. 1 and 2. First, the robot computes the SEIF prediction stage. We assume static beacons and a robot with non-linear kinematic model. Thus, its Jacobian should be computed at each time, which requires recovering μ. Our scheme uses the efficient algorithm described in [21] for SEIF motion update and state recovery. This algorithm computes the predicted information vector $\bar{\xi}$, the information matrix $\bar{\Omega}$ and also the recovered predicted estimate μ.

Algorithm 1 Summary of operation of the robot

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

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 "SEIF state") **then**
5: Compute ξ_i and Ω_i with MS_i as in (5) and (6)
6: Send ξ_i and Ω_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**

Next, the robot broadcasts an *UpdateReq* message that includes *LNB* and the predicted estimate μ. Transmitting the whole μ is not suitable in cases with large numbers of beacons: it would require broadcasting larger –or several– messages, increasing message losses. Besides, the greater part of μ is not of interest for the beacons in *LNB*. The approach adopted is to transmit only the elements in μ required for each beacon in *LNB*. Each beacon b_i in *LNB* takes range measurements to the robot and to the beacons $b_j \in SA_i$. For integrating them it needs μ_i, μ_r and $\mu_j \; \forall j \in SA_i$. Of course, only estimates of beacons b_j that are at "SEIF stage" are available. Let ev_i be the vector with the estimates required for beacon b_i to perform its SEIF update, defined as $ev_i = \begin{bmatrix} \mu_r & \mu_i & \mu_j \end{bmatrix}^T$.

At the beginning *LNB* is assumed empty. If a beacon b_i that received the *UpdateReq* message does not find itself in *LNB*, it sends a *BeaconDiscovery* message to the robot with the identifiers of the beacons that are within SA_i. The next *UpdateReq* message sent by the robot will include *LNB* updated with b_i and the beacon estimate vector ev_i. When a beacon in *LNB* does not participate in a number of consecutive updates, the robot interprets that it is out of its sensing area and excludes it from *LNB*.

After broadcasting the *UpdateReq* message, the robot starts a phase in which it receives *UpdateResp* messages from beacons. A beacon b_i that received an *UpdateReq* message performs differently depending on whether b_i is at the "Initialization stage" or at the "SEIF stage".

1) Beacon at the SEIF stage: In this case b_i measures the range to the beacons in its sensing range –its measurement set is MS_i– and computes its contribution to the SEIF update integrating each measurement as follows:

$$\xi_i = \sum_{k \in MS_i} H^T_{i,k} R^{-1} [z_{i,k} - h_{i,k}(ev_i) + H_{i,k} ev_i], \quad (5)$$

$$\Omega_i = \sum_{k \in MS_i} H^T_{i,k} R^{-1} H_{i,k}, \quad (6)$$

where $h_{i,k}(ev_i)$ and $H_{i,k}$ stand for the predictions and Jacobians for each measurement in MS_i either robot-beacon measurement –it uses observation modes in (2) and (3)– or inter-beacon measurement –(4). R is the covariance matrix of the measurement noise.

Then, beacon b_i transmits the resulting ξ_i and Ω_i to the robot in an *UpdateResp* message. When a timeout expires the robot reconstructs the updated state (ξ and Ω) adding the predicted $\bar{\xi}$ and $\bar{\Omega}$ to the contributions received from beacons:

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

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

where F_i is the projection matrix that implements operations that allocate ξ_i and Ω_i at the correct entries for ξ and Ω.

2) Beacon at the Initialization stage: In this case beacon b_i creates an $UpdateResp$ message with MS_i and transmits it to the robot, which uses MS_i for the initialization of b_i. When the robot finalizes beacon initialization it incorporates it in $\vec{x}$. In this scheme, shown in Alg. 2, the robot centralizes the initialization of every beacon. We call it **Method1**.

We have also developed **Method2**, in which each beacon computes its own initialization: both SEIF update and beacon initialization are decentralized. b_i integrates MS_i in its own initialization. When the beacon initialization is finished, b_i sends the robot its estimation in an $UpdateResp$ message so that it incorporates the initialized beacon in $\vec{x}$.

A number of beacon initialization mechanisms can be used in the proposed scheme. The selection between $Method1$ or $Method2$ depends on the initialization chosen and the beacons computing capabilities. For instance, with beacons implemented using Wireless Sensor Network technology $Method2$ is suitable in case of using trilateration initialization but would require $Method1$ if using PFs. Beacons implemented with devices such as embedded computers or PDAs could execute $Method2$ with PFs.

Finally, the robot should check if the updated Ω satisfies the sparsification bound. If not, the active beacons with stronger links are selected and the weakest are deactivated as described in [8]. Note that measurements from active and passive beacons are integrated in the update stage. Integrating measurements from a passive beacon gives these nodes the possibility of being selected in the sparsification step.

V. Simulations

This section evaluates the accuracy, computational burden and robustness of the proposed scheme and compares it with existing methods. We assume the robot with differential configuration. Its kinematic model is as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + T_k V_k \sin\theta_{k-1} \\ y_{k-1} + T_k V_k \cos\theta_{k-1} \\ \theta_{k-1} + T_k \alpha_k \end{bmatrix}, \qquad (9)$$

where (x_k, y_k, θ_k) is the robot state, V_k and α_k are the odometry linear and steering velocities and T_k is the differential time between t_k and t_{k-1}. The standard deviation of odometry for V and α of $\sigma_V = 0.001 m/s$ and $\sigma_\alpha = 0.005 rad$.

We assume that each beacon is equipped with a range sensor with 15 meters sensing range and a standard deviation $\sigma_m = 1.2m$. Particle Filters were selected for beacon initialization due to their flexibility and robustness. The PFs were configured with 150 particles. When the first measurement for a beacon is received, its auxiliary PF is initialized deploying particles in an annular distribution of width $w = 4\sigma_m$. The sparsification bound chosen was 10, within the range suggested in [21]. SLAM provides the generated map –and robot location– in a local coordinate frame. To compare with the ground-truth, an affine transform is performed on the final beacon locations, re-aligning the local solution into the same global coordinate frame.

The proposed method has been extensively tested with different scenarios and robot trajectories. As an example Fig. 2 shows the results using the robot odometry from the Plaza dataset of Djugash [22] assuming 50 randomly distributed beacons. The resulting mean error in the robot and beacon location estimations were respectively $0.31m$ and $0.42m$.

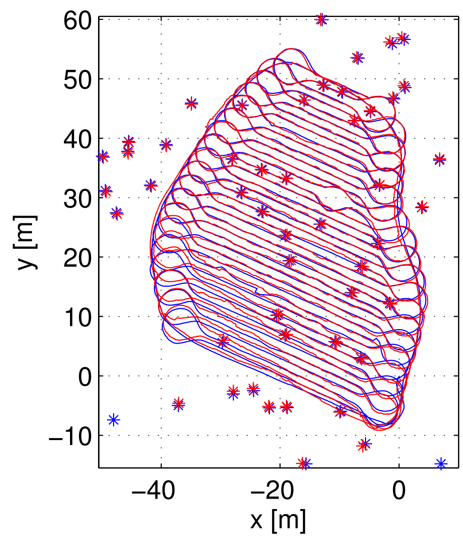

Fig. 2. Results of the proposed *Method1* with 50 randomly distributed beacons. The ground-truth is represented in blue and the estimation in red.

The proposed scheme was compared to EKF and traditional SEIF, in which the robot integrates only robot-beacon measurements. The parameters for all the methods were exactly the same. Series of extensive simulations with different robot motion and random maps were performed. Figure 3 shows the mean error in the estimation of map (top) and robot (center), and the auxiliary PF convergence times (bottom) in 100 $70x70m$ scenarios with 50 beacons. For clarity, only the results for 100 scenarios are shown. In almost all scenarios the proposed *Method2* had the lowest robot and map errors, see Table I. Similar results were obtained with the proposed *Method1*. Also, in the proposed methods PFs converged much sooner than in SEIF and EKF. Earlier beacon initialization has positive effects on errors and on computational burden. These results were confirmed in repetitions with different parameters and measurement and odometry noise levels.

TABLE I

MEAN MAP AND ROBOT ERRORS AND AUXILIARY PF CONVERGENCE TIMES IN SERIES OF 500 SIMULATIONS WITH 50 BEACONS.

	EKF	SEIF	Proposed *Method2*
Map error [m]	0.431	0.451	0.308
Robot error [m]	0.634	0.663	0.415
PF convergence times (k)	1246	1249	263

Series of simulations with different map sizes –between 10 and 300 beacons– were performed to evaluate the scalability of the proposed scheme. 80 scenarios with randomly deployed beacons were simulated for each map size. They employed the same parameters that were used before. Fig-

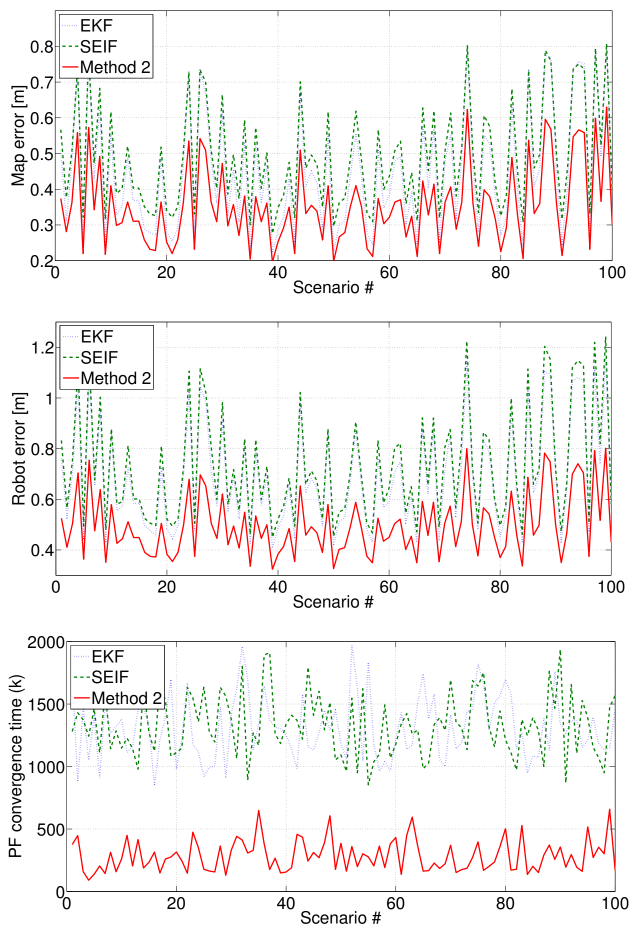

Fig. 3. Comparison of the proposed *Method2*, EKF and traditional SEIF in map error (top), robot error (center) and auxiliary PFs convergence times (bottom) in 100 simulations with different random robot paths and maps. Similar results were obtained with the proposed *Method1*. Results are shown connected for better visualization.

ure 4-top shows the average robot computational burden –measured with *MATLAB Profiler*– for the EKF, traditional SEIF and the proposed *Method1* and *Method2*. *Method2* is the most efficient for maps with more than 50 beacons. For any map size *Method2* requires lower burden than *Method1* and, *Method1* and lower than the traditional *SEIF*.

The map and robot estimation errors obtained in these simulations are in Fig. 4-bottom. The proposed *Method2* obtains the lowest errors. *Method1* and *Method2* obtain similar errors: only *Method2* is shown for clarity. The advantage of the proposed scheme 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 robot odometry noise.

Our scheme uses a protocol between robot and beacons. For realism, in all the previous simulations messages were assumed affected by random transmissions errors. The *Packet Reception Rate* (PRR) model depicted in Fig. 5-top was used. This model is realistic and similar to experimental models such as that in [23]. To evaluate the robustness of our scheme series of experiments have been simulated using the PRR

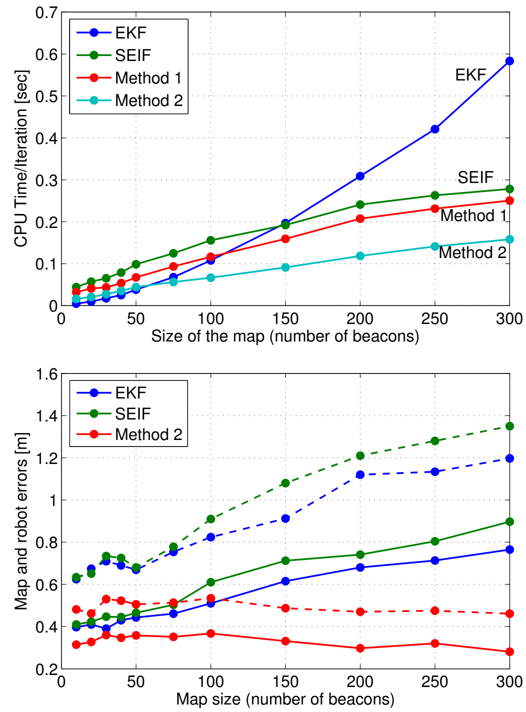

Fig. 4. Performance analysis of the proposed *Method1*, *Method2*, EKF and traditional SEIF with different map sizes: top) robot computational burden, bottom) map and robot errors. Map errors are represented with full lines and robot errors, with dashed lines.

model scaled by a factor in the range [0-1]. Scaling factor 0.5 involves that in the best case (low distances) 50% of the messages are lost. The resulting map errors (full lines) and robot errors (dashed lines) are shown in Fig. 5-bottom. The additive nature of the proposed scheme makes it rather robust to message loss. Figure 5-bottom also shows the errors for EKF and traditional SEIF –horizontal lines since they are not influenced by PRR. The accuracy of the proposed *Method2* overtakes EKF and SEIF with PRR levels higher than 75%, which is a rather pessimistic rate –with low distances 25% of the messages are lost. Also, it is insensitive to transmission loss for PRR levels higher than 90%.

Series of simulations have been performed to assess the impact of different sparsification bounds. Our analysis confirms the results in [21], which have been omitted for brevity.

VI. Experiments

The proposed scheme has been validated in series of experiments performed in the *CONET Integrated Testbed* (`http://conet.us.es`) [4], see Fig. 6-left. The testbed is a remote tool to assess and compare methods combining robot and ubiquitous networks. The testbed is composed of *Pioneer 3-AT* robots, each equipped with a *Hokuyo* range finder, a *Microsoft Kinect*, GPS and Inertial Measurement Units. The robot ground-truth localization is obtained using the AMCL (Adaptive Monte-Carlo Localization) that integrates laser range measurements and provides an error lower than 0.01m. Among others, the testbed also includes a network with 9 *Nanotron NanoPAN* devices equipped with

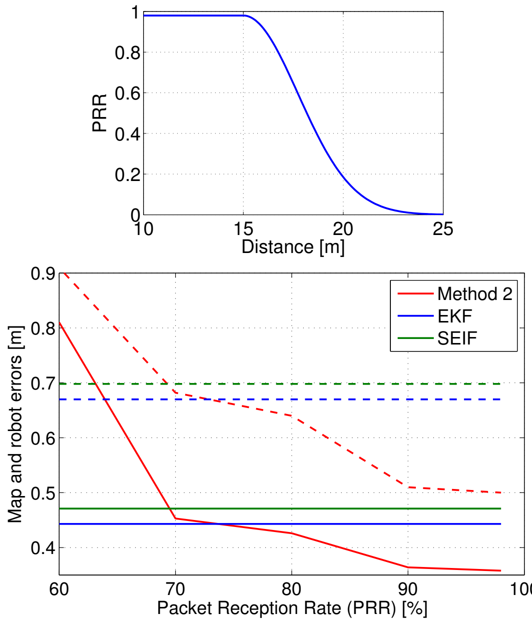

Fig. 5. Top) PRR model used in simulations. Bottom) Impact of PRR on map –full lines– and robot –dashed– errors for the proposed DSEIF *Method2*, EKF and SEIF.

range sensors, Fig. 6-right. Their characterization shown that their error has zero mean and a standard deviation of $\sigma_m = 0.8m$.

Fig. 6. Left) Picture taken in the validation experiments. Right) *Nanotron NanoPAN* range sensors used in the experiments.

Figure 7 shows the results of the proposed *Method2* in one experiment. It was compared with EKF and traditional SEIF in series of 25 experiments with different maps and robot paths, see Table II. The improvement in the map error is 40% w.r.t. SEIF and 37% w.r.t. EKF. The testbed allowed performing the same experiment many times in the same conditions, which confirmed the repeatability of results.

TABLE II

MEAN MAP AND ROBOT ERRORS AND AUXILIARY PF CONVERGENCE TIME IN SERIES OF 25 EXPERIMENTS.

	EKF	SEIF	Proposed *Method2*
Map error [m]	0.251	0.2625	0.1575
Robot error [m]	0.485	0.506	0.425
PF convergence times (k)	1246	1249	263

Figure 8 presents the evolution of the location error for each beacon in the proposed *Method2* (bottom) and the traditional SEIF (top). The drawing for each beacon starts when its auxiliary PF converged. In the proposed method the beacon errors are significantly lower and beacon PFs converge significantly sooner. In average in the traditional SEIF all PFs converged at $k = 1246$ and in our method they converged at $k = 263$ (78% earlier).

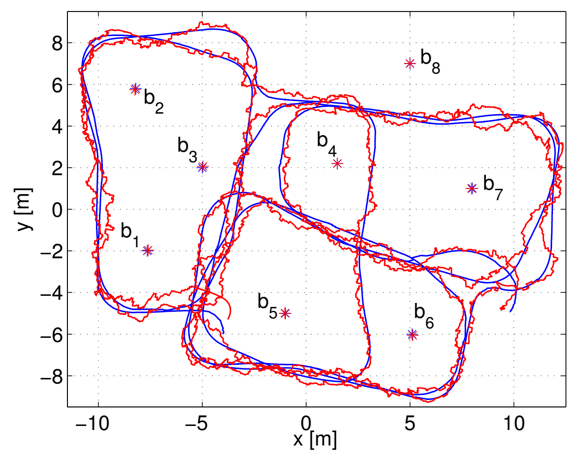

Fig. 7. Results of the proposed *Method2*. Estimations are in red color and ground truth in blue.

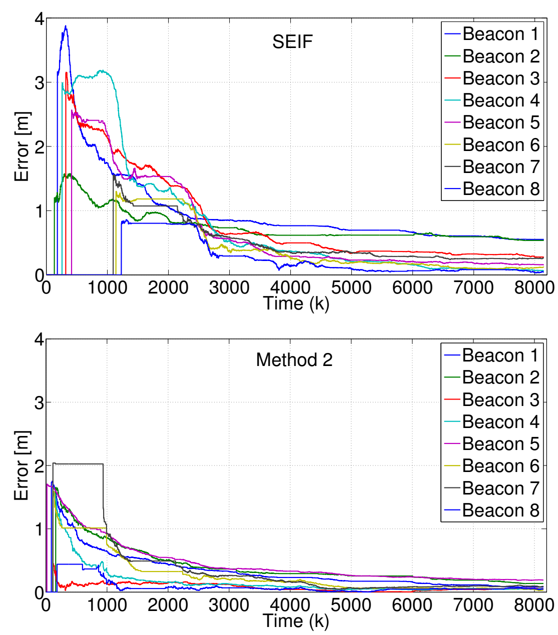

Fig. 8. Evolution of the location error for each beacon for traditional SEIF (top) and the proposed *Method2* (bottom).

In order to evidence the scalability improvement we performed series of hybrid simulated-real experiments with the robot, 8 real beacons and different numbers of simulated beacons at random locations. 25 experiments were performed for each map size with between 20 and 200 beacons. The resulting robot computational burden and errors, in Fig. 9, confirmed the results obtained in simulation.

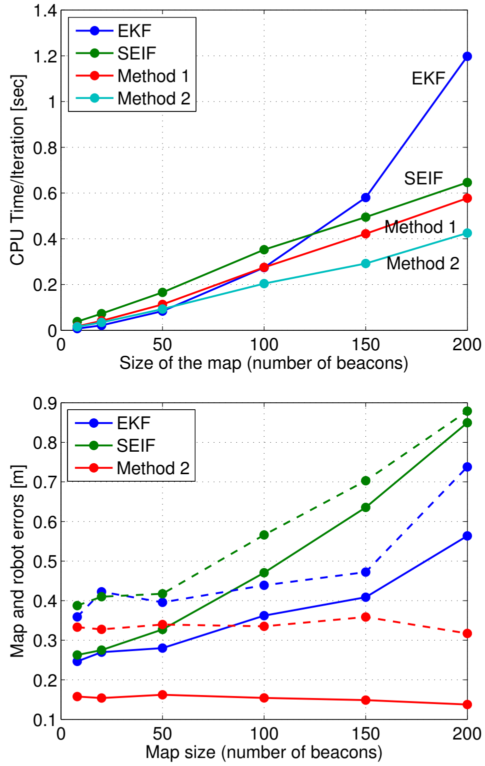

Fig. 9. Performance analysis of the proposed *Method1*, *Method2*, EKF and traditional SEIF with different map sizes: top) robot computational burden, bottom) map and robot errors. Map errors are represented with full lines and robot errors, with dashed lines.

VII. Conclusions

This paper deals with RO-SLAM for robot-sensor network cooperation applications in which sensor nodes (beacons) are used as landmarks for SLAM. This paper proposes a SEIF SLAM scheme that distributes the computation of the SLAM algorithm between the robot and the surrounding beacons exploiting their sensing, computing and communication capabilities. Beacons actively participate in the SLAM computation taking range measurements and integrating them in the SEIF update stage.

The presented scheme naturally integrates robot-beacon measurements together with inter-beacon measurements, which results in improvements in map and robot estimations. Also, it inherits from SEIF its efficiency and scalability and further reduces the robot computational burden by exploiting the beacons computing capability. As a result, the proposed scheme has lower error and lower computer requirements than traditional RO-SLAM methods such as EKF and SEIF. Two different implementations are presented. In *Method1* the initialization of each beacon is centralized at the robot whereas in *Method2* it is performed at each beacon.

The extension of this scheme to multi-robot SLAM harmonizing robot and sensor nodes cooperative behavior and its full-scale experimentation in the real application scenarios are object of current work.

References

[1] S. Challa, F. Leipold, S. Deshpande, and M. Liu, "Simultaneous localization and mapping in wireless sensor networks," in *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005*, dec. 2005, pp. 81 – 87.

[2] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello, "Range-only slam with a mobile robot and a wireless sensor networks," in *IEEE Intl. Conf. on Robotics and Automation, ICRA'09*, 2009, pp. 8–14.

[3] D. Sun, A. Kleiner, and T. Wendt, "Multi-robot range-only slam by active sensor nodes for urban search and rescue," in *RoboCup 2008: Robot Soccer World Cup XII*, ser. Lecture Notes in Computer Science, L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, Eds. Springer Berlin Heidelberg, 2009, vol. 5399, pp. 318–330.

[4] A. Jiménez-González, J. R. Martínez-de Dios, and A. Ollero, "An integrated testbed for cooperative perception with heterogeneous mobile and static sensors," *Sensors*, vol. 11, no. 12, pp. 11 516–11 543, 2011.

[5] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Range-only slam for robots operating cooperatively with sensor networks," in *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA'06*, may 2006, pp. 2078 –2084.

[6] E. Menegatti, M. Danieletto, M. Mina, A. Pretto, A. Bardella, S. Zanconato, P. Zanuttigh, and A. Zanella, "Autonomous discovery, localization and recognition of smart objects through wsn and image features," in *Proc. IEEE GLOBECOM 2010*, 2010, pp. 1653 –1657.

[7] J.-L. Blanco, J.-A. Fernandez-Madrigal, and J. Gonzalez, "Efficient probabilistic range-only slam," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS'08*, sept. 2008, pp. 1017 –1022.

[8] S. Thrun, D. Koller, Z. Ghahmarani, and H. Durrant-Whyte, "Slam updates require constant time," *School of Computer Science, Carnegie Mellon University, Pittsburgh, Tech. Rep*, 2002.

[9] U. Frese, "A proof for the approximate sparsity of slam information matrices," in *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA'05*, 2005, pp. 329–335.

[10] M. A. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Proc. of the 18th Intl. Joint Conference on Artificial Intelligence, IJCAI-03*, G. Gottlob and T. Walsh, Eds. San Francisco, CA: Morgan Kaufmann Publishers, 2003, pp. 1157–1164.

[11] U. Frese, "Treemap: An o (log n) algorithm for simultaneous localization and mapping," in *Spatial Cognition IV. Reasoning, Action, Interaction*. Springer, 2005, pp. 455–477.

[12] Y. Liu and S. Thrun, "Results for outdoor-slam using sparse extended information filters," in *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA'03*, vol. 1, 2003, pp. 1227–1233.

[13] R. Eustice, M. Walter, and J. Leonard, "Sparse extended information filters: Insights into sparsification," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS'05*, 2005, pp. 3281–3288.

[14] S. Thrun and Y. Liu, "Multi-robot slam with sparse extended information filers," in *Robotics Research*. Springer, 2005, pp. 254–266.

[15] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only slam," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS'05*, 2005, pp. 2499–2504.

[16] E. Olson, J. Leonard, and S. Teller, "Robust range-only beacon localization," in *Proc. 2004 IEEE/OES Autonomous Underwater Vehicles*, june 2004, pp. 66 – 75.

[17] M. Banatre, P. Marron, A. Ollero, and A. Wolisz, *Cooperating Embedded Systems and Wireless Sensor Networks*. Wiley, 2008.

[18] D. O. Popa and F. L. Lewis, "Algorithms for robotic deployment of wsn in adaptive sampling applications," in *Wireless Sensor Networks and Applications*. Springer US, 2008, pp. 35–64.

[19] Y. Mei, C. Xian, S. Das, Y. C. Hu, and Y.-H. Lu, "Sensor replacement using mobile robots," *Comput. Commun.*, vol. 30, no. 13, pp. 2615–2626, Sep. 2007.

[20] J. Goerner, N. Chakraborty, and K. Sycara, "Energy efficient data collection with mobile robots in heterogeneous sensor networks," in *IEEE Intl. Conf. on Robotics and Automation, ICRA'13*, 2013.

[21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 3rd ed. Cambridge: The MIT Press, 2005.

[22] J. Djugash, B. Hamner, and S. Roth, "Navigating with ranging radios: Five data sets with ground truth," *Journal of Field Robotics*, vol. 26, no. 9, pp. 689–695, September 2009.

[23] M. Mamun, T. Hasan-Al-Mahmud, S. Debnath, and I. M., "Analyzing the low power wireless links for wireless sensor networks," *Journal of Telecommunications*, vol. 1, pp. 99 –110, February 2010.