An Interlaced Extended Information Filter for Self-Localization in Sensor Networks

Andrea Gasparri and Federica Pascucci

***Abstract*—**

Wireless Sensor Networks (WSN) are at the forefront of emerging technologies due to the recent advances in Micro-Electro-Mechanical Systems (MEMS). The inherent multi-disciplinary nature of WSN attracted scientists coming from different areas stemming from networking to robotics. WSN are considered to be unattended systems with applications ranging from environmental sensing, structural monitoring, and industrial process control to emergency response and mobile target tracking. Most of these applications require basic services such as self-localization or time-synchronization. The distributed nature and the limited hardware capabilities of WSN challenge the development of effective applications. In this paper the Self- Localization problem for Sensor Network is addressed. A distributed formulation based on the Information version of the Kalman Filter is provided. Distribution is achieved by neglecting any coupling factor in the system and assuming an independent reduced-order filter running on-board each node. The formulation is extended by an interlacement technique. It aims to alleviate the error introduced by neglecting the cross-correlation terms by "suitably" increasing the noise covariance matrices. Real experiments involving MICAz Mote platforms produced by Crossbows along with simulations have been carried out to validate the effectiveness of the proposed Self-Localization technique.

***Index Terms*— Sensor Networks, Distributed Applications, Distributed Network**

I. The self-localization problem in Sensor Networks

A sensor network consists of a collection of nodes deployed in an environment that cooperate to perform a task. Each node, which is equipped with a radio transceiver, a micro-controller and a set of sensors, shares data to reach the common objective. Sensor networks provide a framework in which, exploiting the collaborative processing capabilities, several problems can be faced and solved in a new way. However, it comes along with several challenges such as limited processing, storage and communication capabilities as well as limited energy supply and bandwidth. Performing a partial computation locally on each node, and exploiting inter-node cooperation, is the ideal way to use sensor networks. Unfortunately, this *modus-operandi* is highly constrained by the reduced hardware capabilities as well as by the limited energy resources that makes communication very expensive in terms of life-time for a node. As a consequence, these constraints must be taken into account when developing algorithms able to operate in a distributed fashion.

Sensor networks can be of interest to different areas of application, ranging from environmental monitoring [9], [41], civil infrastructures [23], [27], medical care [38], [32] to home and office applications [39], [25]. In each field, the deployment of a sensor network has provided interesting advantages. For instance, in the context of environmental monitor the introduction of a sensor network made it possible to keep environments intrinsically threatening for human beings [41] under surveillance, or in the context of medical care it made it possible to remotely monitor the health condition of patients by continuously extracting clinical relevant information [32].

However, in order to build these application, some basic services, such as time synchronization or nodes localization, are generally required. In fact, basic middle ware services, such as routing, often rely on location information, e.g., geographic routing [5], [40], [24]. Specifically, the localization problem in Sensor Networks consists of finding out the locations of nodes in regards to any topology or metric of interest. This problem turns out to be difficult to solve. In fact in [21], [14] it was proven that a sufficient condition for a sensor network to be localizable cannot be easily identified. This holds even when considering the availability of perfect measurements. Further, several analyses showed that having reliable ranging information is fairly practical [42], [44], [2], especially when using the received signal strength indication (RSSI).

In this paper a distributed formulation based on the Information version of the Kalman Filter is provided to deal with the self localization problem in sensor networks. Distribution is achieved by neglecting any coupling factor in the system and assuming an independent reduced-order filter running on-board each node. The error introduced by this assumption is then mitigated by increasing the noise covariance matrices. This formulation is particularly convenient in all those scenarios where the dimension of the state space is lower than the dimension of the observations. Indeed, this is the case of the proposed sensor network scenario, where the dimension of the state space for each node is equal to two, while the number of the observations is strictly related to the number of nodes deployed into the environment.

The rest of the paper is organized as follows. In Sec. II the state of the art for the localization problem in sensor networks is given. In Sec. III some theoretical insights about the estimation problem in a probabilistic framework are provided. In Sec. IV the interlacement technique is described. In Sec. V the sensor network scenario exploited in this work is detailed. In Sec. VI the formulation of the information filter for the adopted scenario is proposed. In Sec. VII the performance analysis is depicted, while in Sec. VIII the experimental results are described. In Sec. IX the analysis of the computational complexity is detailed. Finally, in Sec. X conclusions are drawn.

II. State of the art

A taxonomy of localization algorithms for sensor networks can be drawn according to the computational organization, i.e., centralized and distributed, to the mechanism adopted for estimating

A. Gasparri and F. Pascucci are with the Department of Computer Science and Automation of the University of Rome "Roma Tre".

location, i.e., range-based or range free, and finally in regards to the availability of anchors nodes, i.e. anchor-based or anchor-free.

Centralized algorithms exploit a central computer to perform all the complex computations using information gathered by nodes [12], [36], [6]. Distributed algorithms dispense the computation over the network, allowing each node to perform locally and compensating for the lack of global knowledge through an intensive collaborative processing [28], [11], [10]. Both schemes offer advantages and drawbacks. Centralized algorithms provide interesting performance but they lack in scalability and robustness. Distributed algorithms provide high robustness and scalability but the development of effective collaborative processing algorithms is challenging.

Range-based algorithms exploit point-to-point distances or angle estimates in order to perform the localization task [33], [37], [30]. Range-free algorithms do not make any assumption about the availability or reliability of this information [20], [26], [44]. Although range-free approaches are appealing as a cost-effective alternative to more expensive range-based approaches, their performance may lack in accuracy.

Anchor-based algorithms rely on the availability of location information for some special nodes in order to localize the network [15], [35]. Anchor-free methods determine the geometry of the network simply by exploiting local interaction among nodes [34], [45]. Anchor-based algorithms have the advantage of directly localizing nodes within a global reference frame, but their accuracy is affected by the number of anchor nodes and their distribution in the sensor field [7]. Conversely, anchor-free methods scale better and do not required expensive hardware, although only relative location estimates can be provided.

Centralized algorithms represent the first attempt to solve the localization problem in sensor networks. In [12], the authors propose the semi-definite programming approach (SDP) to solve the localization problem. The key idea is to model geometric constraints between nodes as linear matrix inequalities (LMIs), then use the semi-definite programming theory to solve it. The result is a bounding region for each node, representing feasible locations where nodes are supposed to be. Although using a set of convex constraints in order to estimate the position of a node is very elegant, it turns out to be inaccurate as constraints do not use precise data range. Moreover, the algorithm provides a good estimation only when having anchors densely deployed on the boundary of the sensor network, a condition that can not always be guaranteed. The SDP approach is extended to deal with noisy distance measurements by taking advantage of an additional technique to mitigate inaccuracies [3]. In fact, the solution provided by the SDP, though not accurate, represents by the authors a good starting point for a gradient-descent method. Furthermore, numerical results show that by means of this improvement it is possible to obtain a solution very close to the optimal one. However, the distributed formulation is the result of a clusterization and a local execution of the algorithm within each subset. Therefore, the computational complexity is merely mitigated reducing the number of nodes but the approach still remains almost centralized. In [36], the authors propose an algorithm that uses connectivity information, i.e., which nodes are within the communication range of which others, to derive the locations of the nodes in the network. This algorithm is based on multidimensional scaling (MDS), a set of data analysis techniques that display the structure of distance-like data as a geometrical picture [4]. It can be broken down into three steps. Starting with the given network connectivity information, an all-pairs shortest-path algorithm is run to roughly estimate the distance between each possible pair of nodes. After, the multidimensional-scaling is applied over these data to derive node locations. Finally, location estimates are normalized with respect to nodes whose position is known.

Distributed algorithms better fit the inherent collaborative nature of sensor networks. In [28], the authors developed an algorithm focused on providing more robust local maps. The idea is to split the problem into a sub-set of smaller regions in which the localization is performed taking advantage of the probabilistic notion of *robust quadrilaterals*. A robust quad is a set of four nodes fully-connected by distance measurements and well-spaced in such a way that no ambiguity can arise, even when in the presence of noise. The algorithm merges the sub-regions using a coordinate system registration procedure. Such a procedure maps local reference systems into a global one providing the best fitting matrix when in presence of a set of common nodes. An optional optimization step can be executed in order to refine the local map first. The weakness of this approach, as pointed out by the same authors, is that under conditions of low node connectivity or high measurement noise, the algorithm may be able to localize only a reduced number of nodes. In [10], the authors propose an approach where localization is performed by exploiting clustering information. Starting from locally-aware anchors, an initial set of calibrated nodes is built. This set is then expanded to include iteratively all the cluster-heads, i.e., the representative node for the cluster. Due to the iterative nature of this approach a refining step is required in order to provide reliable location estimates. Once the cluster-heads have been fully localized, the remaining follower nodes, i.e., non-cluster-head nodes, can be localized.

Range-free algorithms instead may offer an alternative anytime distance-information are not available, due to stringent hardware limitations. In [20], a range-free localization algorithm called APIT is proposed. In this work, the environment is first isolated into triangular regions defined by beacons: localization is achieved by checking whether a node is inside or outside of these regions. Combinations of anchor positions can be used to reduce the diameter of the estimated area. Although an interesting insight on how localization error affects a variety of location-dependent applications such as geographical routing or target tracking is provided, an impractical number of beacons might be required to achieve satisfactory performances. In [44], a sequence-based RF localization algorithm called Ecolocation is proposed. The key idea is to determine the location of unknown nodes by examining the ordered sequence of received signal strength (RSS) measurements taken at multiple reference nodes. The authors propose a constraint-based approach that provides for robust location decoding even in the presence of random RSS fluctuations due to multi-path fading and shadowing. However, the algorithm performance is heavily conditioned by the number of available reference nodes. In [8], the authors propose a RF-based distributed localization method where location are estimated by simply averaging the positions of all the anchors it is connected to. Obviously, the accuracy of the estimation is strictly related to the density of anchors deployed in the environment and the density required to obtain an acceptable estimation is fairly practical.

Anchor-free algorithms may finally represent an alternative solution in case prior knowledge about location are not available and an estimation in regards to a global reference frame is not required. In [34], the authors propose the Anchor-Free Localization algorithm (AFL), an algorithm where all nodes concurrently calculate and refine their coordinate information. The key idea is the introduction of fold-freedom: a fold-free embedding of a graph is an embedding where every cycle has the correct clockwise/counterclockwise orientation of nodes, modulo global reflection, with respect to the original graph. In detail, AFL is composed of two steps. During the first step, a folder-free graph embedding is computed starting from the original embedding and selecting five ad-hoc reference nodes used to approximate the polar coordinate of any other node. Successively, a mass-spring based optimization is performed in order to correct and balance localized errors. In [45], an anchor-free node localization protocol, which exploits clusterization to achieve scalability, is proposed. Such a protocol consists of three steps: network-bootstrapping, local position discovery and global localization. During the first step clusters are identified and a "breadth first spanning tree" rooted at the head of each cluster is performed. Since each node is able to measure distances from its neighbors (by exploiting some TOA technique) and a route exists from it to the cluster headset, all local distance information are sent to the cluster heads. This information will be used during the second step to build a local map at each cluster head. Finally, in the third step cluster heads collaborate in order to obtain a global map of the network. Such a global coordinate system can be built from the local maps by simply exploiting matrix rotations, translations and mirroring.

In this paper, a novel distributed, range-based algorithm, namely the Interlaced Extended Information Filter (IEIF) is proposed. Starting from a centralized formulation, distribution is achieved by neglecting any coupling factor in the system and assuming an independent reduced-order filter running on-board each node. This formulation is successively extended by an interlacement technique aiming to alleviate the error introduced by neglecting the cross-correlation terms by "suitably" increasing the noise covariance matrices. The proposed algorithm can provide global localization by assuming anchors are available. In the same way also relative localization among nodes can be achieved by relaxing the assumption of anchors availability. The effectiveness of this distributed approach has been thoroughly investigated by experiments carried out with MICAz Mote platforms produced by Crossbows, while its scalability has been analyzed by means of simulations.

III. Theoretical Background

A. Bayesian framework

The probability theory provides a suitable framework for modeling the Self-Localization problem in Sensor Networks. Let us consider a system described by the following set of equations:

$$\begin{aligned} x_k &= f(x_{k-1}, u_k, w_k) \\ z_k &= h(x_k, v_k) \end{aligned} \tag{1}$$

where x_k is a stochastic variable representing the locations of the nodes, u_k is the control input, w_k and v_k are noises that affect the system, while $f(\cdot)$ and $h(\cdot)$ are mathematical relations that characterize the state transition and the observation z_k respectively.

In the probabilistic context, the localization problem consists of computing the probability distribution $p(x_k|Z_k, U_k)$ for all

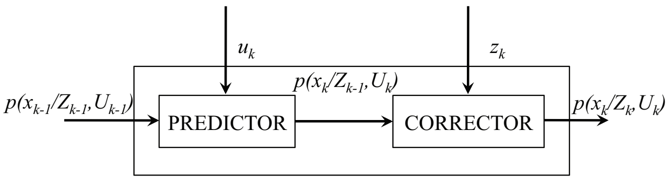


Fig. 1. Bayesian filter

times k. This probability distribution describes the *joint posterior density* of the sensor locations (x_k) given the recorded observations (Z_k) and control inputs (U_k) up to time k. To apply this approach in a real context, it is often required to perform the above mentioned computation online. Therefore, a recursive formulation should be provided in terms of Bayesian filter, graphically depicted in Fig. 1.

The idea is to provide at each time step k a new estimate by combining the available estimate of the *joint posterior distribution* $p(x_{k-1}|Z_{k-1}, U_{k-1})$ at time $k-1$, with the control u_k and the observation z_k. In this way, both the *state transition model* and the *observation model*, describing respectively the stochastic effects of the control input and observation, are required.

From a probabilistic point of view, the state transition model can be described in terms of the *joint prior density* $p(x_k|x_{k-1}, u_k)$. Such probability distribution exploits that the state transition is assumed to be a Markov process in which the next state x_k depends only on the immediately preceding state x_{k-1} and the applied control u_k and is independent of the observations.

On the other hand, the observation model describes the probability of retrieving an observation z_k when the sensor locations are known, and is generally stated in the form $p(z_k|x_k)$.

The localization algorithm can be implemented in a standard two-step recursive prediction (*time–update*)

$$p(x_k|Z_{k-1}, U_k) = \int_{\Xi} p(x_k|x_{k-1}, u_k) p(x_{k-1}|Z_{k-1}, U_{k-1}) dx_{k-1} \tag{2}$$

and correction (*measurement update*) form

$$p(x_k|Z_k, U_k) = \frac{p(z_k|x_k) p(x_k|z_{k-1}, U_k)}{p(z_k|Z_{k-1}, U_k)}. \tag{3}$$

Equations (2) and (3) provide a recursive procedure for calculating the joint posterior $p(x_k|Z_k, U_k)$, however they cannot be implemented on a digital computer in their general form stated above, as the joint posterior over the state space is a density over a continuous space, hence has infinitely many dimensions. Therefore, any effective localization algorithm has to resort to additional assumptions.

B. The Kalman Filter

A common approach is represented by the use of Kalman filter [22]. In this context a linear or linearized system model is required

$$\begin{aligned} x_k &= F_k x_{k-1} + B_k u_k + w_k \\ z_k &= H_k x_k + v_k \end{aligned} \tag{4}$$

where $w_k \sim \mathcal{N}(0, Q_k)$, $v_k \sim \mathcal{N}(0, R_k)$, $x_0 \sim \mathcal{N}(\hat{x}_0, P_0)$ are mutually independent Gaussian variables for each pair of time instant (k, k'). The joint posterior $p(x_k|Z_k, U_k)$ is modeled by a unimodal Gaussian density. The mode of this density ($\hat{x}_k$) yields the current positions of the nodes, and the variance (P_k) represents the current uncertainty. As only these two parameters have to be computed to propagate uncertainty, there is no need to discretize the state space. In this way the prediction becomes

$$\begin{array}{rcl} \hat{x}_{k|k-1} & = & F_k\hat{x}_{k-1|k-1} + Bu_k \\ P_{k|k-1} & = & F_k P_{k-1|k-1} F_k^T + Q_k \end{array} \tag{5}$$

while the correction requires the computation of the well known Kalman gain matrix

$$K_k = P_{k|k-1} H_k^T \left[H_k P_{k|k-1} H_k^T + R_k \right]^{-1} \tag{6}$$

before update the estimate

$$\begin{array}{rcl} \hat{x}_{k|k} & = & \hat{x}_{k|k-1} + K_k(z_k - H_k\hat{x}_{k|k-1}) \\ P_{k|k} & = & P_{k|k-1} - K_k\left[H_k P_{k|k-1} H_k^T + R_k\right] K_k^T. \end{array} \tag{7}$$

The advantage of Kalman filter lies in its efficiency and in the high accuracy that can be obtained, however it is not able to cope with high nonlinear system and multimodal distributions. Therefore in most practical situation, Kalman filter cannot be applied. Instead, one is forced to use approximations or suboptimal solutions. Over the years a large number of approximate nonlinear filters has been proposed in literature [13]. Some are fairly general, while others are more tailored to a particular application.

Here only analytic approximations have been considered: in this category, it is included the Extended Kalman Filter (EKF). The main feature of this filter is that it linearizes the non linear functions in the state transition and observation models. The EKF is derived for non linear systems with additive noise

$$\begin{array}{rcl} x_k & = & f(x_{k-1}, u_k) + w_k \\ z_k & = & h(x_k) + v_k \end{array} \tag{8}$$

where w_k and v_k are mutually independent, zero mean white Gaussian random sequences, having covariance matrices Q_k and R_k respectively. The nonlinear functions $f(\cdot)$ and $h(\cdot)$ are approximated by the first term in their Taylor series expansion. The joint posterior density is approximated by a Gaussian density and computed recursively as follows

- Prediction

$$\begin{array}{rcl} \hat{x}_{k|k-1} & = & f(\hat{x}_{k-1|k-1}, u_k) \\ P_{k|k-1} & = & J_x^f P_{k-1|k-1} {J_x^f}^T + Q_k \end{array} \tag{9}$$

- Update

$$\begin{array}{rcl} K_k & = & P_{k|k-1} {J_x^h}^T \left[J_x^h P_{k|k-1} {J_x^h}^T + R_k \right]^{-1} \\ \hat{x}_{k|k} & = & \hat{x}_{k|k-1} + K_k\left(z_k - h(\hat{x}_{k|k-1})\right) \\ P_{k|k} & = & P_{k|k-1} - K_k\left[J_x^h P_{k|k-1} {J_x^h}^T + R_k\right] K_k^T \end{array} \tag{10}$$

where J_x^f and J_x^h is the Jacobian of the nonlinear functions $f(\cdot)$ and $h(\cdot)$ respectively.

As only Kalman filters are used in the sequel, only these techniques have been reported for sake of space, however a complete review can be found in [13].

C. The Information Filter

An Information Filter (IF) is essentially a Kalman Filter (KF) expressed in terms of measures of *information* about the parameters (state) of interest rather than direct state estimates and their associated covariances [19]. The two key information-analytic variables are the *information matrix* and the *information state vector*, where the term information is used according to the Fisher definition.

The Fisher information matrix Ψ_k is the amount of information that an observable random variable z carries about an unobservable parameter x upon which the likelihood function of z, $L(x) = p(z \mid x)$, depends. It can be derived as the covariance of the *score function*, that is the partial derivative, with respect to some parameter x, of the logarithm (commonly the natural logarithm) of the likelihood function. If the observation is z and its likelihood is $L(x) = p(z \mid x)$, then the score $\mathcal{S}_k(x)$ can be described as follows:

$$\begin{array}{rcll} \mathcal{S}_k(x) & = & \nabla_x \ln p(z_k \mid x_k) & (11) \\ & = & \dfrac{\nabla_x p(z_k \mid x_k)}{p(z_k \mid x_k)}. & (12) \end{array}$$

Moreover, being the expectation of the score:

$$\begin{array}{rcll} E[\mathcal{S}_k(x)] & = & \displaystyle\int \frac{\nabla_x p(z_k \mid x_k)}{p(z_k \mid x_k)} p(z_k \mid x_k)\, dz_k & (13) \\ & = & \nabla_x \displaystyle\int p(z_k \mid x_k)\, dz_k & (14) \\ & = & \nabla_x 1 = 0 & (15) \end{array}$$

the Information matrix Ψ_k is simply the second order moment of the score function $\mathcal{S}_k(x)$, as follows:

$$\begin{array}{rcll} \Psi_k & = & E[\mathcal{S}_k(x)\mathcal{S}_k(x)^T] & (16) \\ & = & E[\{\nabla_x \ln p(z_k \mid x_k)\}\{\nabla_x \ln p(z_k \mid x_k)\}^T]. & (17) \end{array}$$

Furthermore, if the following regularity condition holds:

$$\int H_x(p(z_k \mid x_k)) = \nabla_x \nabla_x^T p(z_k \mid x_k) = 0 \tag{18}$$

where H_x is the square matrix of second-order partial derivatives (i.e., Hessian Matrix), the Information matrix Ψ_k can be also written as:

$$\Psi_k = -E[\nabla_x \nabla_x^T \ln p(z_k \mid x_k)] \tag{19}$$

At this point, when the likelihood function $p(z \mid x)$ is a Gaussian distribution and the posterior conditional distribution is Gaussian as well, described as $p(x_k \mid z) \sim \mathcal{N}(\hat{x}_k, P_k)$, then it can be proved [29] that the Information Matrix is equal to the inverse of the covariance matrix P_k as follows

$$\Psi_k = P_k^{-1}. \tag{20}$$

Likewise, the information state vector y_k can be easily derived as the product of the inverse of the information matrix and the state estimate as follows:

$$\begin{array}{rcll} y_k & = & \Psi_k\, x_k & (21) \\ & = & P_k^{-1}\, x_k. & (22) \end{array}$$

The Information Filter formulation can be easily derived from the Kalman Filter formulation under the assumption of Gaussianity previously stated. In particular, by performing the substitutions given in eq. (22) and eq. (20), the following set of equations is obtained:

- Prediction

$$\begin{array}{rcl} \Psi_{k|k-1} & = & \left[F_k(\Psi_{k-1|k-1})^{-1} F_k^T + Q_k\right]^{-1} \\ L_{k|k-1} & = & \Psi_{k|k-1} F_k \Psi_{k-1|k-1}^{-1} \\ \hat{y}_{k|k-1} & = & L_{k|k-1}\hat{y}_{k-1|k-1} + \Psi_{k|k-1} B_k u_k \end{array} \tag{23}$$

- Estimation

$$\begin{aligned} \Psi_{k|k} &= \Psi_{k|k-1} + \Phi_k \\ \hat{y}_{k|k} &= \hat{y}_{k|k-1} + i_k \\ \Phi_k &= H_k^T R_k^{-1} H_k \\ i_k &= H_k^T R_k^{-1} z_k. \end{aligned} \tag{24}$$

The information filter can be extended to a linearized estimation algorithm for nonlinear system, the Extended Information Filter (EIF). The idea is to apply the analytic approximations used in EKF and the substitutions of IF to build up an estimation method for nonlinear systems. The EIF presents several interesting features, among the others an easy initialization of matrices and vectors, a reduced computational load, and aptitude to be distributed for parallel computation. The EIF equations can be found as follows

- Prediction

$$\begin{aligned} \Psi_{k|k-1} &= \left[J_x^f (\Psi_{k-1|k-1})^{-1} {J_x^f}^T + Q_k\right]^{-1} \\ \hat{y}_{k|k-1} &= \Psi_{k|k-1} f(\hat{x}_{k-1|k-1}, u_k) \end{aligned}$$

- Estimation

$$\begin{aligned} \Psi_{k|k} &= \Psi_{k|k-1} + \Phi_k \\ \hat{y}_{k|k} &= \hat{y}_{k|k-1} + i_k \\ \Phi_k &= {J_x^h}^T R_k^{-1} J_x^h \\ i_k &= {J_x^h}^T R^{-1}\left[z_k - h(\hat{x}_{k|k-1}) + J_x^h \hat{x}_{k|k-1}\right]. \end{aligned} \tag{25}$$

A more comprehensive description of the Information Filter derivation is given in [29].

IV. On the interlacement of EKF and EIF

The interlacement technique has been developed [18] to reduce the computational load of a nonlinear filter by means of splitting the estimation of the state variables into parallel subfilters. The key idea is derived from the multi-players dynamic game theory, where each player chooses its own strategy as the optimal response to the strategy adopted by the other players. In the framework of estimation, players are represented by subfilters, strategy by estimate, whereas the optimal response depends on the estimation algorithm. The interlacement technique can be applied both to the Extended Kalman Filter and the Extended Information Filter, as detailed below.

A. Interlaced Extended Kalman Filter

The Interlaced Extended Kalman Filter (IEKF) has been introduced to distribute the estimate of an EKF over a network of processors, each one devoted to estimate a subspace of the state variables minimizing the loss of cross-correlation links. For sake of clarity, let us consider a system whose model can be decomposed into 2 subsystems and rewritten as (for the first filter $i = 1$ and $j = 2$, while for the second $i = 2$ and $j = 1$)

$$\begin{aligned} x_k^{(i)} &= f^{(i)}(x_{k-1}^{(i)}, x_{k-1}^{(j)}, u_k) + w_k^{(i)} \\ z_k^{(i)} &= h^{(i)}(x_k^{(i)}, x_k^{(j)}) + v_k^{(i)}. \end{aligned} \tag{26}$$

The IEKF equations proceed from EKF filter equations (see Fig. 2). At the k-th step, each subfilter form a prediction

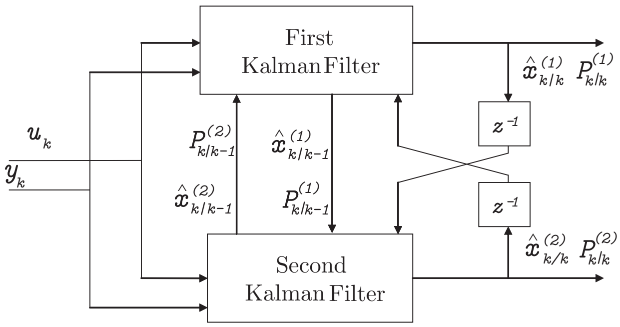


Fig. 2. Interlaced Kalman Filter

exploiting both its own estimation and the one of the other filter, according with the following equation

$$\hat{x}_{k|k-1}^{(i)} = f^{(i)}(\hat{x}_{k-1|k-1}^{(i)}, \hat{x}_{k-1|k-1}^{(j)}, u_{k-1}) \tag{27}$$

$$P_{k|k-1}^{(i)} = J_{x,i}^{f,i} P_{k-1|k-1}^{(i)} {J_{x,i}^{f,i}}^T + \tilde{Q}_k^{(i)} \tag{28}$$

where

$$\tilde{Q}_k^{(i)} = Q_k^{(i)} + J_{x,j}^{f,i} P_{k-1|k-1}^{(j)} {J_{x,j}^{f,i}}^T \tag{29}$$

being $J_{x,i}^{f,i}$ and $J_{x,j}^{f,i}$ the Jacobians of the relation $f^{(i)}(\cdot)$ with respect to $x_k^{(i)}$ and $x_k^{(j)}$.

After prediction step the estimates elaborated by the two subfilters are exchanged and used during the update step.

In this step the observation prediction is formed and compared with the measure z_k provided by the system

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_k^{(i)}[z_k - h^{(i)}(\hat{x}_{k|k-1}^{(i)}, \hat{x}_{k|k-1}^{(j)})] \tag{30}$$

$$P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} J_{x,i}^{h,i} P_{k|k-1}^{(i)} \tag{31}$$

where the Kalman gain is computed applying the relation

$$K_k^{(i)} = P_{k|k-1}^{(i)} {J_{x,i}^{h,i}}^T [J_{x,i}^{h,i} P_{k|k-1}^{(i)} {J_{x,i}^{h,i}}^T + \tilde{R}_k^{(i)}]^{-1}$$

in which

$$\tilde{R}_k^{(i)} = R_k + J_{x,j}^{h,i} P_{k|k-1}^{(j)} {J_{x,j}^{h,i}}^T \tag{32}$$

where $J_{x,j}^{h,i}$ and $J_{x,j}^{h,i}$ are the Jacobians of $h^{(i)}(\cdot)$ with respect to $x_k^{(i)}$ and $x_k^{(j)}$.

From (29) and (32) it can be noticed that the process and measurement noise covariance matrices $Q_k^{(i)}$ and $R_k^{(i)}$ are suitable increased by addition of positive semi definite quantities that take into account the error introduced by the decoupling operation. It is easy to recognize that the term added to $R_k^{(i)}$ in (32) represents the cross-correlation between the filters due to innovation process, while the term added to $Q_k^{(i)}$ in (29) is related to the cross-correlation induced by propagation process.

B. Interlaced Extended Information Filter

The Interlaced Extended Information Filter (IEIF) is the counterpart of the IEKF in the information space. As already mentioned, the IEF is suitable for distributing the estimation process over parallel computation units, due to the loose correlation between the elements of the information vector. In the information space, indeed, the correlation between information variables, that are not explicitly connected or directly involved in a measurement, is not represented, whereas the covariance matrix explicitly stores this relation in the corresponding off-diagonal entries. Apart from this rough correlation between information variables, there is still a coupling factor that has to be taken into account even in presence of distributed implementation of EIF to prevent the divergence of the filter itself. In order to consider this coupling factor, the Interlaced Extended Information Filter is introduced in this work. Let us consider again a system whose model can be decomposed into 2 subsystems having model equations expressed by (26). The IEIF schema is represented in Fig. 2, after substituting IEKF with IEIF while recalling the relation given by (20). At each time the filter compute a prediction and estimation step exploiting the equations below

- Prediction

$$\begin{array}{rcl} \Psi^{(i)}_{k|k-1} &=& \left[J^{f,i}_{x,i}(\Psi^{(i)}_{k-1|k-1})^{-1}{J^{f,i}_{x,i}}^T + \tilde{Q}_k\right]^{-1} \\ \hat{y}^{(i)}_{k|k-1} &=& \Psi_{k|k-1} f(\hat{x}^{(i)}_{k-1|k-1}, \hat{x}^{(j)}_{k-1|k-1}, u_k) \\ \tilde{Q}^{(i)}_k &=& Q^{(i)}_k + J^{f,i}_{x,j}\left(\Psi^{(j)}_{k-1|k-1}\right)^{-1}{J^{f,i}_{x,j}}^T \end{array} \quad (33)$$

- Estimation

$$\begin{array}{rcl} \Psi^{(i)}_{k|k} &=& \Psi^{(i)}_{k|k-1} + \Phi^{(i)}_k \\ \hat{y}^{(i)}_{k|k} &=& \hat{y}^{(i)}_{k|k-1} + i^{(i)}_k \\ \Phi^{(i)}_k &=& {J^{h,i}_{x,i}}^T \tilde{R}^{(i)-1}_k J^{h,i}_{x,i} \\ i^{(i)}_k &=& {J^{h,i}_{x,i}}^T \tilde{R}^{(i)-1}_k z^{(i)\prime}_k \\ \tilde{R}^{(i)}_k &=& R^{(i)}_k + J^{h,i}_{x,j}\left(\Psi^{(j)}_{k|k-1}\right)^{-1}{J^{h,i}_{x,j}}^T \\ z^{(i)\prime}_k &=& \nu^{(i)}_k + J^{h,i}_{x,i}\hat{x}^{(i)}_{k|k-1} + J^{h,i}_{x,j}\hat{x}^{(j)}_{k|k-1} \\ \nu^{(i)}_k &=& z^{(i)}_k - h^{(i)}(\hat{x}^{(i)}_{k|k-1}, \hat{x}^{(j)}_{k|k-1}). \end{array} \quad (34)$$

After every single step the sub-filters exchange their results in terms of best estimate and the associate covariance. The estimate is used to compute the expected measurement, whereas the covariance matrix is involved in the computation of the matrices $\tilde{Q}_k$ and $\tilde{R}_k$. These matrices have the same meaning introduced for IEKF and convey the coupling factor between information variables in subsystems i and j.

V. Sensor Network Scenario

In this paper, a group of Ω nodes deployed on a planar environment is considered. A typical sensor network node's hardware consists of a microprocessor with reduced computational capability, a radio component, several sensor devices, a minimal data storage unit and a battery with limited life. Furthermore, a few nodes are equipped with an absolute position system device so that localization in regards to a global frame can be obtained for the whole network. Finally, nodes are assumed to be motionless.

The state of the node i at time k is described by its location with respect to a global frame as follows:

$$x^{(i)}_k = [\,p^{(i)}_{x,k}\; p^{(i)}_{y,k}\,]^T \quad (35)$$

Thus the state of the whole system is the vector obtained by collecting the locations of all nodes:

$$x_k = [x^{(1)T}_k, \cdots, x^{(\Omega)T}_k]^T \quad (36)$$

A. System model

Since nodes are assumed to be still, the model of the i-th node is simply given by:

$$x^{(i)}_k = x^{(i)}_{k-1} + w^{(i)}_k \quad (37)$$

where $w^{(i)}_k \in \mathbb{R}^2$ is a zero mean white noise vector with covariance matrix $Q^{(i)}_k$.

Note that, the system is naturally fully decoupled as the state transition of a node does not depend upon other nodes. This property turns out to be very useful for the distributed formulation of the filter. Furthermore, the framework allows to mix static nodes with mobiles one simply by changing the state transition model according to the kinematics of each sensor node [31].

B. Observation model

Nodes are equipped with several sensor devices. In particular, a way to measure inter-node distances is assumed to be available. The related observation model can be obtained considering the Euclidian distance as follows:

$$\begin{array}{rcll} z^{(i,j)}_k &=& h^{(i,j)}\big(x^{(i)}_k, x^{(j)}_k\big) + v^{(i)}_k & (38) \\ &=& \parallel x^{(i)}_k - x^{(j)}_k \parallel + v^{(i)}_k & (39) \\ &=& \sqrt{\big(p^{(i)}_{x,k} - p^{(j)}_{x,k}\big)^2 + \big(p^{(i)}_{y,k} - p^{(j)}_{y,k}\big)^2} + v^{(i)}_k & (40) \end{array}$$

where $v^{(i)}_k \in \mathbb{R}$ is a zero mean white noise vector with covariance $R^{(i)}_k$.

VI. The Information Filter for Sensor Networks

Due to the nonlinear nature of the observation model, the linear Information Filter previously introduced cannot be applied as it is. An extension to deal with the non-linearity of the observation model is required. Note that having a linear prediction model results in a "hybrid" Information Filter: with the prediction equation of a linear IF and the estimation equation of an Extended IF. In the following, a centralized formulation of the filter is proposed. Then, a distributed one based on simplifying assumptions is devised. However, both filters can be summarized by the same two-stage formulation:

- Prediction

$$\begin{array}{rcl} \Psi_{k|k-1} &=& [\Psi^{-1}_{k-1|k-1} + Q_k]^{-1} \\ L_{k|k-1} &=& \Psi_{k|k-1}\Psi^{-1}_{k-1|k-1} \\ \hat{y}_{k|k-1} &=& L_{k|k-1}\hat{y}_{k-1|k-1} \end{array} \quad (41)$$

- Estimation

$$\begin{array}{rcl} \Psi_{k|k} &=& \Psi_{k|k-1} + \Phi_k \\ \hat{y}_{k|k} &=& \hat{y}_{k|k-1} + i_k \\ \Phi_k &=& {J^h_x}^T R^{-1}_k J^h_x \\ i_k &=& {J^h_x}^T R^{-1}_k z'_k \\ z'_k &=& \nu_k + J^h_x \hat{x}_{k|k-1} \\ \nu_k &=& z_k - h(\hat{x}_{k|k-1}) \end{array} \quad (42)$$

Differences between the centralized formulation and the distributed formulation are merely related to the state space dimension and to the construction of the Jacobian matrix J^h_x.

A. Centralized EIF

In the case of the centralized formulation, the whole state of the system as given in eq. (36) is considered. Therefore, the computation of the complete Jacobian matrix involves all the inter-distance measurements available over the network. In particular, given a generic observation $z_k^{(i,j)}$, representing the distance from the node i to the node j measured by the node i, the related Jacobian row is:

$$J^{h(i,j)} = \begin{bmatrix} \mathbf{0} & J_{x,i}^{h(i,j)} & \mathbf{0} & J_{x,j}^{h(i,j)} & \mathbf{0} \end{bmatrix} \tag{43}$$

where

$$J_{x,i}^{h(i,j)} = \begin{bmatrix} \frac{p_x^{(i)} - p_x^{(j)}}{d} & \frac{p_y^{(i)} - p_y^{(j)}}{d} \end{bmatrix} = -J_{x,j}^{h(i,j)} \tag{44}$$

with

$$d = \sqrt{\left(p_x^{(i)} - p_x^{(j)}\right)^2 + \left(p_y^{(i)} - p_y^{(j)}\right)^2} \tag{45}$$

According to this notation, given the following set of observations $\mathbf{z_k} = \left[z_k^{(i,j)}, z_k^{(l,j)}, z_k^{(i,l)}, z_k^{(l,i)}\right]^T$ among three nodes $\{x_i, x_j, x_l\}$, the resulting Jacobian matrix J_x^h is:

$$\begin{aligned} J_x^h &= \begin{bmatrix} J^{h(i,j)^T} & J^{h(l,j)^T} & J^{h(i,l)^T} & J^{h(l,i)^T} \end{bmatrix}^T \\ &= \begin{bmatrix} \mathbf{0} & J_{x,i}^{h(i,j)} & \mathbf{0} & J_{x,j}^{h(i,j)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & J_{x,l}^{h(l,j)} & J_{x,j}^{h(l,j)} & \mathbf{0} \\ \mathbf{0} & J_{x,i}^{h(i,l)} & J_{x,l}^{h(i,l)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & J_{x,i}^{h(l,i)} & J_{x,l}^{h(l,i)} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{aligned} \tag{46}$$

B. Distributed EIF

A distributed formulation can be introduced by means of some simplifying assumptions. The system model is linear and fully decoupled thus suitable for a distributed implementation, while the Jacobian matrix J_x^h features some couplings. In particular, for each node i the following Jacobian block $J_x^{h(i)}$ can be considered:

$$J_x^{h(i)} = \begin{bmatrix} \mathbf{0} & J_{x,i}^{h(i,j)} & \mathbf{0} & J_{x,j}^{h(i,j)} & \mathbf{0} \\ \mathbf{0} & J_{x,i}^{h(i,l)} & J_{x,l}^{h(i,l)} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{47}$$

Furthermore, according to eq. (47), it can be noticed that if a node i considers its neighbors as anchors at each time-step, the partial derivatives of node j are always naughts for a generic Jacobian row $J_x^{h(i)}$. Therefore, the related Jacobian block $J_x^{h(i)}$ becomes:

$$\begin{aligned} J_x^{h(i)} &= \begin{bmatrix} \mathbf{0} & J_{x,i}^{h(i,j)} & \mathbf{0} & J_{x,j}^{h(i,j)} & \mathbf{0} \\ \mathbf{0} & J_{x,i}^{h(i,l)} & J_{x,l}^{h(i,l)} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \left[\begin{array}{c|c|ccc} \mathbf{0} & J_{x,i}^{h(i,j)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & J_{x,i}^{h(i,l)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] \\ &= \left[\begin{array}{c|c|ccc} \mathbf{0} & J_{x,i}^{h,i} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] \end{aligned}$$

In this way the complete Jacobian matrix J_x^h, described in eq. (46), turns out to be a block-matrix. Therefore, the centralized formulation can be easily decomposed in a set of Ω reduced-order filters, each one run by a single node with the aim of estimating its location with respect to information (in terms of observations and latest estimates) coming from the other nodes.

Furthermore, the capability of the algorithm to perform the localization process with or without anchors can be explained by the fact that for how the algorithm is conceived, neighbors are always considered as anchors. Thus, the availability of real anchors does not affects the formulation, but the accuracy of the localization process

C. Interlaced EIF

Considering the neighborhood of each node as a set of anchors helps to distributed the formulation of the Extended Information Filter. However, at the same time an error is introduced into the estimation process as a consequence of this approximation. The Interlacement technique introduced in Sec. IV-A turns out to be an effective solution to mitigate the error introduced by this simplifying assumption. The resulting formulation for the sensor network scenario is

- Prediction

$$\begin{aligned} \Psi_{k|k-1}^{(i)} &= \left[\left(\Psi_{k-1|k-1}^{(i)}\right)^{-1} + Q_k^{(i)}\right]^{-1} \\ L_{k|k-1}^{(i)} &= \Psi_{k|k-1}^{(i)}\left(\Psi_{k-1|k-1}^{(i)}\right)^{-1} \\ \hat{y}_{k|k-1}^{(i)} &= L_{k|k-1}^{(i)}\hat{y}_{k-1|k-1}^{(i)} \end{aligned} \tag{48}$$

- Estimation

$$\begin{aligned} \Psi_{k|k}^{(i)} &= \Psi_{k|k-1}^{(i)} + \Phi_k^{(i)} \\ \hat{y}_{k|k}^{(i)} &= \hat{y}_{k|k-1}^{(i)} + i_k^{(i)} \\ \Phi_k^{(i)} &= J_{x,i}^{h,i\,T}\tilde{R}_k^{(i)^{-1}} J_{x,i}^{h,i} \\ i_k^{(i)} &= J_{x,i}^{h,i\,T}\tilde{R}_k^{(i)^{-1}} z_k^{(i)'} \\ \tilde{R}_k^{(i)} &= R_k^{(i)} + \sum_{j\in\mathcal{N}(i)} J_{x,j}^{h,i}\left(\Psi_{k|k-1}^{(j)}\right)^{-1} J_{x,j}^{h,i\,T} \\ z_k^{(i)'} &= \nu_k + J_{x,i}^{h,i}\hat{x}_{k|k-1}^{(i)} \\ \nu_k^{(i)} &= z_k^{(i)} - h^{(i)}(\hat{x}_{k|k-1}^{(i)}, \xi_k^{(i)}) \end{aligned} \tag{49}$$

where $\xi_k^{(i)} = \left\{\hat{x}_{k|k-1}^{(j)} : j \in \mathcal{N}_k(i)\right\}$. Note that, the interlacement contribution does not add any significant complexity to the estimation process as the Jacobian term $J_{x,j}^{h,i}$ is simply obtained by negation of the term $J_{x,i}^{h,j}$ and the term $\left(\Psi_{k|k-1}^{(j)}\right)^{-1}$ is broadcasted by the neighbors.

D. Algorithmic Derivation

From an algorithmic point of view, a possible implementation of the distributed EIF running onboard each node is given in algorithm (1). In detail, at each iteration k a given node i performs the following four steps: it listens for a pre-fixed τ amount of time waiting for new data $\left\{z_k^{(i)}, \Psi_k^{(i)}\right\}$ broadcasted by each other node within its range of visibility, successively it updates its estimate $x_k^{(i)}$ by executing the set of equations given in eq. (48) and eq. (49) where the Jacobian matrix $Jh^{(i)}$ is built, as previously described in eq. (48), according to the collected data $\left\{z_k^{(i)}, \Psi_k^{(i)}\right\}$, finally it notifies to the network its latest estimate $\left(y_{k|k}^{(i)}, \Psi_{k|k}^{(i)}\right)$. Note that, no clock synchronization is required for the sensor network, indeed the temporal index k for the data coming from neighboring nodes is simply meant as the most recent available so far.

Algorithm 1: Reduced-Order Filter

Data: $\{\hat{y}^{(i)}_{k-1|k-1}, \Psi^{(i)}_{k-1|k-1}\}$

Result: $\{\hat{y}^{(i)}_{k|k}, \Psi^{(i)}_{k|k}\}$

```
/* Data Collecting                                          */
```

$\{z^{(i)}_k, \Psi^{(i)}_k\} \leftarrow$ listening_procedure(τ) ;

where:

$z^{(i)}_k = \{z^{(i,j_1)}_k, \cdots, z^{(i,j_{M_i})}_k\},$

$\Psi^{(i)}_k = \{\hat{y}^{(j_1)}_{k-1|k-1}, (\Psi^{(j_1)}_{k-1|k-1})^{-1}, \cdots, \hat{y}^{(j_{M_i})}_{k-1|k-1}, (\Psi^{(j_{M_i})}_{k-1|k-1})^{-1}\}$

```
/* Updating Step                                            */
```

$\{\hat{y}^{(i)}_{k|k-1}, \Psi^{(i)}_{k|k-1}\} \leftarrow$ prediction_proc$(\hat{y}^{(i)}_{k-1|k-1}, \Psi^{(i)}_{k-1|k-1})$;

```
/* Estimation Step                                          */
```

$\{\hat{y}^{(i)}_{k|k}, \Psi^{(i)}_{k|k}\} \leftarrow$ estimation_procedure$(z^{(i)}_k, \Psi^{(i)}_k)$

```
/* Notification Step                                        */
```

notification_procedure$(\hat{y}^{(i)}_{k|k}, \Psi^{(i)}_{k|k})$

VII. PERFORMANCE ANALYSIS

Several computer simulations have been performed in order to investigate the effectiveness of the proposed distributed interlaced extended information filter on a large scale. Moreover, a comparison with an interlaced extended Kalman filter has been carried out as well. Note that, in this work the attention is focused on the design of an interlacement technique within the information filtering. For this reason only a comparison between distributed version of the algorithms is provided. The reader is referred to the works [16], [17] for a detailed analysis concerning a comparison with the centralized versions of the algorithms.

In particular, the following aspects of interest have been considered:

- Level of noise of observations,
- Scalability of the algorithm,

while the following two indexes of quality have been used:

- Estimation accuracy,
- Convergence velocity.

The former is given in terms of distance between the estimated and the real location of a node. The Euclidian distance is adopted as metric. Maximum, minimum and average errors computed over the whole network are considered. The latter is given in terms of number of steps required by the algorithm to settle around the best estimation. This index provides an evaluation of the "reactivity" of the algorithm.

Table I describes how the performance indexes vary with respect to different levels of noise. Convergence is assumed to be reached when the fluctuation of the estimate was bounded within a predefined interval, ± 0.5 cm. A configuration involving 90 nodes with 30 anchors deployed on a 30 m $\times$ 30 m environment is considered. According to the obtained results, the accuracy of the estimation is considerably influenced by the level of noise, while it does not seem to significantly affect neither the convergence time nor the percentage of failures, i.e., the number of unsuccessful trials, of the algorithm.

TABLE I
STATISTICAL ANALYSIS: INDEXES OF QUALITY.

Conf	Noise Std 0.05 [m]		Noise Std 0.2 [m]		Noise Std 0.5 [m]	
	IEIF	IEKF	IEIF	IEKF	IEIF	IEKF
Max Error [cm]	2.88	1.79	11.64	6.97	28.52	18.12
Min Error [cm]	0.14	0.06	0.56	0.24	1.47	0.62
Mean Error [cm]	1.09	0.57	4.38	2.30	11.09	5.92
Std. Dev. [cm]	0.60	0.36	2.40	1.44	6.00	3.77
Conv. Step	21	39	21	42	25	45
% Failure	3	10	4	3	4	4

Table II investigates the scalability of the proposed algorithm. In this analysis, the level of noise is fixed to 0.1 for all the configurations. Furthermore, anchors are supposed to be uniformly distributed and both the environmental size and the ration between the number of nodes and number of anchors are kept constant. According to the obtained results, the higher is the number of nodes the quicker is the convergence of the algorithm. In the same way, the number of failures decreases with an increasing number of nodes. On the other hand, the accuracy of the estimation is not significantly influenced by the dimension of the sensor network, being this related to the hardware sensing capabilities.

TABLE II
STATISTICAL ANALYSIS: INDEXES OF QUALITY.

Conf	N:120 A:30		N:180 A:45		N:360 A:90	
	IEIF	IEKF	IEIF	IEKF	IEIF	IEKF
Max Error [cm]	5.59	3.53	4.23	3.18	4.11	3.01
Min Error [cm]	0.27	0.11	0.13	0.07	0.11	0.05
Mean Error [cm]	2.17	1.16	1.52	0.96	1.30	0.8
Conv. Step	20	45	17	31	15	27
% Failure	4	6	1	0	0	0

The algorithm might converge to alternative admissible solutions. Indeed, given Ω nodes with Θ anchors, "symmetrical" solutions may exist in regard to the deployment of the anchors. In the case of perfect communication, i.e., no packet is lost, and fully connectivity among nodes, placing anchors on the boundary is a sufficient condition to have an unique solution. However, in a real system some nodes may not be able to communicate with other nodes, therefore in practice alternative plausible solutions with respect to the available data may always exist.

VIII. EXPERIMENTAL RESULTS

Experimental results have been performed to validate the proposed distributed interlaced extended information filter in a real context. In particular, apart from the IEKF, a comparison against a third algorithm, i.e., the ESDP algorithm, has been considered. In detail, the ESDP algorithm is a semi-definite programming (SDP) relaxation approach proposed in [43], for which the code is freely available at `http://www.stanford.edu/~yyye/`.

The network is composed of MICAz (MPR2400) platform, a generation of Motes from Crossbow Technology. The MPR2400 (2400 MHz to 2483.5 MHz band) uses the Chipcon

CC2420, IEEE 802.15.4 compliant, ZigBee ready radio frequency transceiver integrated with an Atmega128L micro-controller. It provides also a flash serial memory, as well as a 51 pin I/O connector that allows several sensor and data acquiring boards to be connected to it. MICAz platform comes along with TinyOS, an open-source event-driven operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS component library includes network protocols, distributed services, sensor drivers, and data acquisition tools, all of which can be used as–is or be further refined for a custom application.

A. Ranging Technique

A ranging technique based on the Time of Arrival (ToA) principle is exploited to compute inter-node distances. The implementation consists of a node sending first a RF packet and emitting an acoustic signal right after. For the receiving node, the RF signal, whose propagation can be assumed instantaneous, is used to trigger a timer, while the acoustic signal, whose propagation delay is measurable, is used to stop it. By the measurement of such a propagation delay and by knowing the propagation velocity of the acoustic signal, the distance is then computed.

Regarding the MICAz platform, the MTS300 and the MTS310 boards, both providing a sounder and a microphone, have been exploited. The sounder is a simple 4 kHz fixed frequency piezoelectric resonator, while the microphone can be used either for acoustic ranging or for general acoustic recording and measurement. Therefore, the RF and acoustic (sounder) signals are exploited for the implementation of the proposed ranging technique.

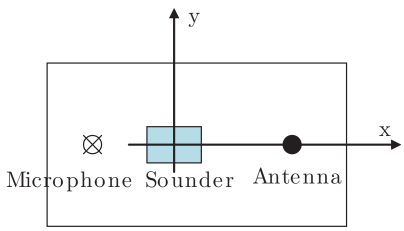
The proposed ranging technique for MICAz platforms has been thoroughly investigated in order to determine the achievable performance. A significant amount of inter-node distances (more than 200 measurements) were collected and a statistical analysis was performed. A precision of $3 \sim 8$ cm with a standard deviation of $8 \sim 14$ cm was experienced considering distances ranging from 20 cm to 2.5 m.

In addition, experiments have been carried out to verify whether the mutual orientation of nodes might influence the measured distance. For such a reason, two nodes have been arranged on the floor at the distance of 54 cm from each other. Such a distance has been manually measured from the sounder of the emitter to the microphone of the receiver. Successively, data has been collected considering different orientations of nodes, in order to simulate a realistic random deployment on the ground. Table III shows the statistic results using again more than 200 measurements for each configuration. According to this analysis differential mutual orientations do not significantly influence the measure of distances. However, as mentioned above, data presents a bias as well as a considerable standard deviation that makes their use challenging.

The bias and the standard deviation describe the uncertainty in the observing process. Several are the sources of such uncertainty. First of all, the parameters used to characterize the propagation velocity of an acoustic wave in the air have been considered fixed, while they change according with humidity and temperature. Secondly, the transmission protocol introduces a delay, which cannot be taken into account, as it is not directly observable.

Exp.	Mean value [m]	Std dev [m]	node 1 orientation [rad]	node 2 orientation [rad]
1	0.5781	0.1229	$\pi/2$	$3\pi/2$
2	0.5734	0.1331	$3\pi/2$	0
3	0.5888	0.1146	$3\pi/2$	$3\pi/2$
4	0.5696	0.1052	$3\pi/2$	π
5	0.5933	0.1098	$3\pi/2$	$\pi/2$
6	0.6008	0.1230	$5\pi/4$	$3\pi/4$
7	0.5972	0.1217	$5\pi/4$	$\pi/4$
8	0.5853	0.1136	$5\pi/4$	$5\pi/4$
9	0.5683	0.1181	$5\pi/4$	$5\pi/2$
10	0.5892	0.1186	$5\pi/4$	π
11	0.5786	0.1239	$5\pi/4$	$7\pi/4$
12	0.5668	0.1299	0	0

TABLE III
INTER-NODE RANGING TECHNIQUE: EXPERIMENTAL RESULTS.



B. Deployment and Evaluation

Several network deployments have been considered for the algorithm evaluation. Each deployment has been obtained by taking advantage of the regularity of the flooring grid and real locations were manually measured exploiting such a regularity. Note that, the extent of the deployment region has been constrained by the hardware capability of the MICAz nodes. Indeed, experiments reveals measurements to be sufficiently reliable only within a range of approximately 2 meters. Here, results related only to two configurations are reported.

Fig. 3 shows the first deployment where 10 nodes are considered. Each node is ideally within the communication range of each other. This way a full connected graph is achieved.

Table IV describes the result of the experiment involving the first environment (Fig. 3). Three different arrangements of anchors are considered, each one involving three nodes. According to experimental results for this configuration, varying the set of anchors does not significantly influence the accuracy of estimation. In particular, the ESDP algorithm has proven to perform slightly better for all the anchors configurations. This can be explained by the fact that being the ESDP a centralized algorithm, it can take advantage of the full availability of information. Nevertheless, it is worthy to mention that a similar estimation accuracy is achieved also by the other two algorithms despite from the partial information availability. In regard to the convergence velocity, the EIF turns out to perform slightly better. This can be explained by the fact that while the Kalman filter requires an initialization for the covariance matrix P which slows down the convergence, the Information Filter does not require any initialization for the information matrix [29].

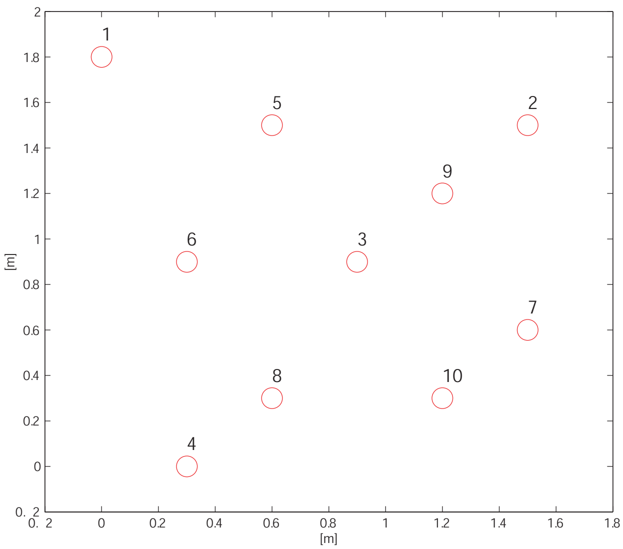


Fig. 3. First deployment: 3 (variable) anchors - 7 nodes.

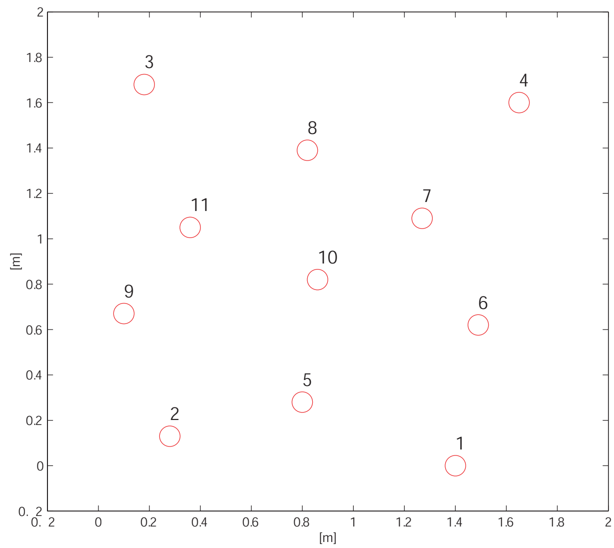


Fig. 4. Second deployment: 3 (variable) anchors - 8 nodes.

Interlaced Extended Information Filter				
Anchors	Max Error [cm]	Min Error [cm]	Mean Error [cm]	Conv. [Steps]
{1,2,3}	12.0	2.9	6.4	8
{1,6,7}	11.8	4.1	6.7	14
{5,9,10}	12.8	1.3	5.2	12

Interlaced Extended Kalman Filter				
Anchors	Max Error [cm]	Min Error [cm]	Mean Error [cm]	Conv. [Steps]
{1,2,3}	11.1	4.1	7.9	25
{1,6,7}	11.0	2.5	6.5	26
{5,9,10}	12.7	1.9	5.6	24

Edge-based SDP relaxation (ESDP)				
Anchors	Max Error [cm]	Min Error [cm]	Mean Error [cm]	Conv. [Steps]
{1,2,3}	7.3	1.6	5.5	-
{1,6,7}	11.7	2.6	6.5	-
{5,9,10}	11.6	2.5	6.8	-

TABLE IV
FIRST DEPLOYMENT: IEIF VS IEKF VS ESDP.

Interlaced Extended Information Filter				
Anchors	Max Error [cm]	Min Error [cm]	Mean Error [cm]	Conv. [Steps]
{1,2,4}	21.0	1.4	12.9	23
{3,4,5}	18.7	6.3	11.8	17
{5,6,11}	22.4	8.6	16.6	25

Interlaced Extended Kalman Filter				
Anchors	Max Error [cm]	Min Error [cm]	Mean Error [cm]	Conv. [Steps]
{1,2,4}	23.6	9.4	15.9	22
{3,4,5}	16.5	10.1	13.2	17
{5,6,11}	21.9	8.9	17.4	39

Edge-based SDP relaxation (ESDP)				
Anchors	Max Error [cm]	Min Error [cm]	Mean Error [cm]	Conv. [Steps]
{1,2,3}	19.2	2.3	11.4	-
{3,4,5}	17.4	2.6	11.7	-
{5,6,11}	20.6	2.3	11.6	-

TABLE V
SECOND DEPLOYMENT: IEIF VS IEKF VS ESDP.

Fig. 4 shows the second deployment where 11 nodes are considered. Again, each node is ideally within the communication range of each other in order to have a full connected graph. However, it should be pointed out that at each iteration only a portion of the network is able to collaborate, due to the high number of outliers occurring in the measurement process. This implies that only partial information is available to the nodes.

Table V describes the result of the experiment involving the second environment (Fig. 4). Also in this deployment, three different arrangements of anchors are considered, each one involving three nodes. According to experimental results for this configuration, varying the set of anchors still does not significantly influences the performance. However, a general deterioration of the estimation accuracy can be noticed, due to the large number of packets lost which considerably reduces the amount of information available for localization. Furthermore, also in this scenario the ESDP algorithm has proven to be perform slightly better . In particular, the advantages related to the availability of full information are more evident in this scenario where a significant lost of packets was experienced.

IX. COMPUTATIONAL COMPLEXITY ANALYSIS

Here, a comparative analysis regarding the computational complexity of both the Interlaced Extended Information Filter previously introduced and the Interlaced Extended Kalman Filter given in Sec. IV-A is propsed. In order to achieve that, the asymptotic notation (a mathematical notation used to describe the asymptotic behavior of functions) is considered. Its purpose is to characterize a function behavior for very large (or very small) inputs in a simple but rigorous way that enables comparison to other functions [1].

Furthermore, in order to easily analyze the filter equations, a formalism has been introduced with the aim of describing the matrix operations and the related computational complexity:

- `SUM(NxM,NxM)` $= \mathcal{O}(N \cdot M)$
- `SUB(NxM,NxM)` $= \mathcal{O}(N \cdot M)$
- `MUL(NxM,MxP)` $= \mathcal{O}(N \cdot M \cdot P)$
- `INV(NxN)` $= \mathcal{O}(N^3)$

Note that, for sake of simplicity, the asymptotic complexity assumed for these operations does not reflect the most efficient implementation available so far. However, it does not affect the validity of the analysis since the complexity of the most efficient implementations scale approximatively the same. Furthermore, all the elementary operations related to scalar values have been assumed with complexity $\mathcal{O}(1)$.

A. The Interlaced Extended Kalman Filter

The complexity of the Interlaced Extended Kalman Filter running on-board of a node can be summarized as in Table VI, where N is the dimension of each node state and M is the number of measurements (neighbors) for each node.

TABLE VI
INTERLACED EXTENDED KALMAN FILTER COMPUTATIONAL LOAD

$\hat{x}^{(i)}_{k\|k-1} = \hat{x}^{(i)}_{k-1\|k-1}$	-
$P^{(i)}_{k\|k-1} = [P^{(i)}_{k-1\|k-1} + Q^{(i)}_k]$	`SUM(NxN,NxN)`
$\hat{x}^{(i)}_{k\|k} = x^{(i)}_{k\|k-1} + K^{(i)}_k \nu_k$	`SUM(Nx1,Nx1)` `MUL(NxM,Mx1)`
$\nu^{(i)}_k = z^{(i)}_k - h^{(i)}(\hat{x}^{(i)}_{k\|k-1}, \xi^{(i)}_k)$	`SUB(Mx1,Mx1)`

$K^{(i)}_k = P^{(i)}_{k\|k-1} J^{h,i^T}_x (S^{(i)}_k)^{-1}$	`MUL(NxN,NxM)`
	`MUL(NxM,MxM)`
	`INV(MxM)`

$S^{(i)}_k = J^{h,i}_x P_{k\|k-1} J^{h,i^T}_x + \tilde{R}^{(i)}_k$	`MUL(MxN,NxM)` `SUM(MxM,MxM)`

$P^{(i)}_{k\|k} = (I - K^{(i)}_k J^{h,i}_x) P^{(i)}_{k\|k-1}$	`MUL(NxM,MxN)`
	`SUB(NxN,NxN)`
	`MUL(NxN,NxN)`

$\tilde{R}^{(i)}_k = R^{(i)}_k + \sum_{j \in \mathcal{N}(i)} J^{h,i}_{x,j} P^{(j)}_{k\|k-1} J^{h,i^T}_{x,j}$	`SUM(MxM,MxM)`
	`MUL(1xN,NxN)`
	`MUL(1xN,Nx1)`
	`M times`

Three remarks are now in order:

- Only matrix operations have been taken into account,
- The complexity of the Jacobian construction has been neglected,
- The complexity of the observation evaluation has been neglected.

The first observation underlines that the asymptotic behavior of the algorithm is desired. The second observation comes from the consideration that the computational complexity of the Jacobian is always lighter compared to other operations. Thus, it will be omitted for sake of clarity. The third observation follows the same reasoning as the second one.

B. The Interlaced Extended Information Filter

The complexity of the distributed Extended Information Filter running on-board of a node can be summarized as in Table VII. The same considerations that have been done for the IEKF still hold here.

TABLE VII
EXTENDED INFORMATION FILTER COMPUTATIONAL LOAD

$\Psi^{(i)}_{k\|k-1} = [(\Psi^{(i)}_{k-1\|k-1})^{-1} + Q^{(i)}_k]^{-1}$	`INV(NxN)`
	`SUM(NxN,NxN)`
	`INV(NxN)`

$L^{(i)}_{k\|k-1} = \Psi^{(i)}_{k\|k-1} (\Psi^{(i)}_{k-1\|k-1})^{-1}$	`MUL(NxN,NxN)`
$\hat{y}^{(i)}_{k\|k-1} = L^{(i)}_{k\|k-1} \hat{y}^{(i)}_{k-1\|k-1}$	`MUL(NxN,Nx1)`
$\Psi^{(i)}_{k\|k} = \Psi^{(i)}_{k\|k-1} + \Phi^{(i)}_k$	`SUM(NxN,NxN)`
$\Phi^{(i)}_k = J^{h,i^T}_x (\tilde{R}^{(i)}_k)^{-1} J^{h,i}_x$	`INV(MxM)` (diag) `MUL(NxM,MxM)` (diag) `MUL(NxM,MxN)`
$\hat{y}^{(i)}_{k\|k} = \hat{y}^{(i)}_{k\|k-1} + i^{(i)}_k$	`SUM(Nx1,Nx1)`
$i^{(i)}_k = J^{h,i^T}_x (\tilde{R}^{(i)}_k)^{-1} z^{(i)'}_k$	`MUL(NxM,Mx1)`

$z^{(i)'}_k = \nu^{(i)}_k + J^{h,i}_{x,i} \hat{x}^{(i)}_{k\|k-1}$	`MUL(MxN,Nx1)`
	`SUM(Mx1,Mx1)`
	`INV(NxN)`
	`MUL(NxN,Nx1)`

$\nu^{(i)}_k = z^{(i)}_k - h^{(i)}(\hat{x}^{(i)}_{k\|k-1}, \xi^{(i)}_k)$	`SUB(Mx1,Mx1)`

$\tilde{R}^{(i)}_k = R^{(i)}_k + \sum_{j \in \mathcal{N}(i)} J^{h,i}_{x,j} (\Psi^{(j)}_{k\|k-1})^{-1} J^{h,i^T}_{x,j}$	`SUM(MxM,MxM)`
	`MUL(1xN,NxN)`
	`MUL(1xN,Nx1)`
	`M times`

C. IEKF vs. IEIF

In order to find out the differences between the two algorithms, the matrix operations have been compared:

TABLE VIII
COMPUTATIONAL COMPLEXITY: COMPARATIVE TABLE I

IEKF	EIF
`SUM(NxN,NxN)`	`INV(NxN)`
`SUM(Nx1,Nx1)`	`SUM(NxN,NxN)`
`MUL(NxM,Mx1)`	`INV(NxN)`
`SUB(Mx1,Mx1)`	`MUL(NxN,NxN)`
`MUL(NxN,NxM)`	`MUL(NxN,Nx1)`
`MUL(NxM,MxM)`	`SUM(NxN,NxN)`
`INV(MxM)`	`INV(MxM)` (diag)
`MUL(MxN,NxM)`	`MUL(NxM,MxM)` (diag)
`SUM(MxM,MxM)`	`MUL(NxM,MxN)`
`MUL(NxM,MxN)`	`SUM(Nx1,Nx1)`
`SUB(NxN,NxN)`	`MUL(NxM,Mx1)`
`MUL(NxN,NxN)`	`MUL(MxN,Nx1)`
`SUM(MxM,MxM)`	`SUM(Mx1,Mx1)`
`MUL(1xN,NxN)xM`	`INV(NxN)`
`MUL(1xN,Nx1)xM`	`MUL(NxN,Nx1)`
	`SUB(Mx1,Mx1)`
	`MUL(1xN,Nx1)`
	`MUL(1xN,NxN)xM`
	`MUL(1xN,Nx1)xM`

Table VIII, which summarizes the set of operations required by both algorithms at each iteration, can be simplified considering that from an asymptotical standpoint, some operations, such as sum, subtraction or transposition, have a lower order than other ones, such as multiplication or inversion.

Table IX can be further simplified considering that from an asymptotical point of view, the number of occurrences, if not related to any of the parameters of interest, does not influence the complexity of the algorithm.

Table X describes the subset of operations characterizing the

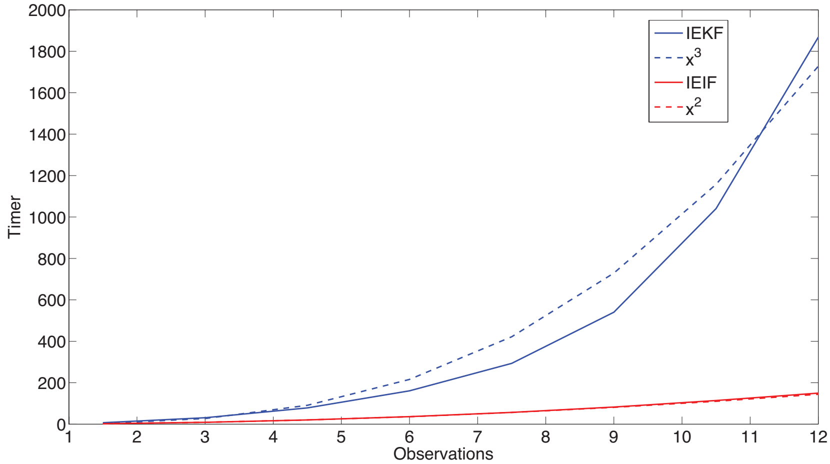


Fig. 5. Computational load: IEIF vs IEKF.

TABLE IX
COMPUTATIONAL COMPLEXITY: COMPARATIVE TABLE II

IEKF	A.C.C.	EIF	A.C.C.
		INV(NxN)	$\mathcal{O}(N^3)$
MUL(NxM,Mx1)	$\mathcal{O}(N \cdot M)$	INV(NxN)	$\mathcal{O}(N^3)$
		MUL(NxN,NxN)	$\mathcal{O}(N^3)$
MUL(NxN,NxM)	$\mathcal{O}(N^2 \cdot M)$	MUL(NxN,Nx1)	$\mathcal{O}(N^2)$
MUL(NxM,MxM)	$\mathcal{O}(N \cdot M^2)$		
INV(MxM)	$\mathcal{O}(M^3)$	INV(MxM) (diag)	$\mathcal{O}(M)$
MUL(MxN,NxM)	$\mathcal{O}(N \cdot M^2)$	MUL(NxM,MxM) (diag)	$\mathcal{O}(N \cdot M)$
		MUL(NxM,MxN)	$\mathcal{O}(N^2 \cdot M)$
MUL(NxM,MxN)	$\mathcal{O}(N^2 \cdot M)$		
		MUL(NxM,Mx1)	$\mathcal{O}(N \cdot M)$
MUL(NxN,NxN)	$\mathcal{O}(N^3)$	MUL(MxN,Nx1)	$\mathcal{O}(M \cdot N)$
MUL(1xN,NxN)xM	$\mathcal{O}(N^2 \cdot M)$		
MUL(1xN,Nx1)xM	$\mathcal{O}(N \cdot M)$	INV(NxN)	$\mathcal{O}(N^3)$
		MUL(NxN,Nx1)	$\mathcal{O}(N^2)$
		MUL(1xN,NxN)xM	$\mathcal{O}(N^2 \cdot M)$
		MUL(1xN,Nx1)xM	$\mathcal{O}(N \cdot M)$

TABLE X
COMPUTATIONAL COMPLEXITY: COMPARATIVE TABLE III

IEKF	A.C.C.	EIF	A.C.C.
MUL(NxM,Mx1)	$\mathcal{O}(N \cdot M)$	INV(NxN)	$\mathcal{O}(N^3)$
		MUL(NxN,NxN)	$\mathcal{O}(N^3)$
MUL(NxN,NxM)	$\mathcal{O}(N^2 \cdot M)$	MUL(NxN,Nx1)	$\mathcal{O}(N^2)$
INV(MxM)	$\mathcal{O}(M^3)$		
MUL(MxN,NxM)	$\mathcal{O}(N \cdot M^2)$		
		MUL(NxM,MxN)	$\mathcal{O}(N^2 \cdot M)$
MUL(NxN,NxN)	$\mathcal{O}(N^3)$	MUL(MxN,Nx1)	$\mathcal{O}(M \cdot N)$
		MUL(NxN,Nx1)	$\mathcal{O}(N^2)$

computational complexity of the two approaches. The dominant operation for the IEIF can be either the multiplication of a matrix $N \times M$ with a matrix $M \times N$ with complexity $\mathcal{O}(N^2 \cdot M)$ or the inversion of a matrix $N \times N$ with complexity $\mathcal{O}(N^3)$, where N is the dimension of the state space and M is the number of observations. Conversely, for the IEKF the dominant operation can be either the inversion of a matrix $N \times N$ with complexity $\mathcal{O}(N^3)$ or the inversion of a matrix $M \times M$ with complexity $\mathcal{O}(M^3)$.

The use of one technique over the other depends upon the reciprocal dimension between the state space and the observations. If the dimension of the state space is lower than the dimension of the observations $N < M$, the IEIF turns out to be computationally more efficient than the IEKF. Conversely, if the dimension of the state space is higher than the dimension of the observations $N > M$, the IEKF performs better even though the complexity is the same from an asymptotical point of view. Indeed, this is due to the fact that several operations with cubic complexity in N are required by the IEIF at each iteration. Note that for the proposed Sensor Network scenario, the dimension of the state space for each node is fixed to $N = 2$, while the dimension of the observations is strictly related to the number of nodes Ω deployed into the environment. Therefore the Interlaced Extended Information Filter turns out to be more effective than the Interlaced Extended Kalman Filter. The same considerations would apply even if a 3-dimensional scenario ($N = 3$) for deployment were considered. Fig. 5 shows the computational load for the two algorithms with respect to an increasing number of nodes. Note that, in this analysis the dimension of the state space for each node was fixed to $N = 2$, the ratio between the number of nodes and the number of anchors was kept constant and so was the size of the environment. In this way, despite the random nature of the deployment, the number of observations available for each node was increasing with respect to an increasing number of nodes. According to the results given in Table X, the computational load for the EIKF clearly shows a cubic trend while the computational load for the IEIF shows a quadratic trend.

Special case: single observation update: Thus far an analysis where M observations were processed all together at each iteration has been provided. The computational complexity can be even further reduced if considering a single observation at a time. However it should be noticed that this solution has two major drawbacks: the convergence time is significantly increased and the accuracy of the estimation can be highly affected by the order in which the measurement are processed, due to the nonlinear nature of the observations. In this case, for the IEKF the inversion of the innovation is reduced to the inversion of a scalar. The dominant operation is given by the multiplication required for the computation of this scalar and its complexity becomes linear with the dimension of the state. However, since it has to be repeated M times the real complexity becomes $\mathcal{O}(N \cdot M)$, which is indeed significantly lower compared to the previous one ($\mathcal{O}(M^3)$). Note that, the situation for the EIF is completely different. In fact, even if the computational load required for the construction of the Innovation matrix becomes linear with the dimension of the state, several inversions of matrixes $N \times N$ are still required at each iteration. Therefore in this case any potential advantage simply vanishes.

X. Conclusions

In this paper, the Interlaced Extended Information Filter (IEIF) for self-localization in Sensor Network has been introduced. The centralized formulation has been distributed by neglecting any coupling factor in the system and assuming an independent reduced-order filter running on-board each node. The original formulation has been successively extended by an interlacement technique which aims to alleviate the error introduced by neglecting the cross-correlation terms by "suitably" increasing the noise covariance matrices.

The effectiveness of the proposed formulation has been investigated via both computer simulations and real experiments involving the MICAz mote platforms produced by Crossbows. In addition, a comparison with an Interlaced Extend Kalman Filter (IEKF) has been provided.

Computer simulations focused on investigating the efficacy of the proposed algorithm in a large scale. The obtained results evidence comparable performance underlining the algebraic equivalence of the two approaches. Experimental results focused on investigating the effectiveness of the proposed algorithm in a real scenario. Also in this case, the obtained results evidence comparable performance . However, according to the performed computational complexity analysis, the IEIF outperforms the IEKF anytime the dimension of the state state space is lower than the dimension of the observations ($N < M$). Indeed, this is the case of the proposed sensor network scenario, where the dimension of the state space for each node is fixed to $N = 2$, while the dimension of the observations is strictly related to the number of nodes Ω deployed into the environment. Finally, the same considerations would apply even if a 3-dimensional state space were considered.

References

[1] Jeremy Avigad and Kevin Donnelly. Formalizing o notation in isabelle/hol. In *Automated Reasoning: Second International Joint Conference, 2004. IJCAR 2004.*

[2] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM*, volume 2, pages 775–784, 2000.

[3] P. Biswas, T. C. Liang, K. C. Toh, Y. Ye and T. C. Wang. Semidefinite Programming Approaches for Sensor Network Localization With Noisy Distance Measurements. In *Automation Science and Engineering, IEEE Transactions on.*, 4(3):360–371, 2006.

[4] Ingwer Borg and Patrick Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag New York, Inc., first edition edition, 1996. (Springer Series in Statistics).

[5] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[6] J. Bruck, J. Gao, and A. A. Jiang. Localization and routing in sensor networks by local angle information. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 181–192, New York, NY, USA, 2005. ACM Press.

[7] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann. Scalable coordination for wireless sensor networks: Self-configuring localization systems. In *Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA '01)*. University of California, Los Angeles, July 2001.

[8] Nirupama Bulusu, Vladimir Bychkovskiy, Deborah Estrin, and John Heidemann. Scalable, ad hoc deployable, rf-based localization. In *Proceedings of the Grace Hopper Celebration of Women in Computing*, Vancouver, British Columbia, Canada, October 2002.

[9] A. Cerpa, J. Elson, M. Hamilton, J. Zhao, D. Estrin, and L. Girod. Habitat monitoring: application driver for wireless communications technology. In *SIGCOMM LA '01: Workshop on Data communication in Latin America and the Caribbean*, pages 20–41, New York, NY, USA, 2001. ACM.

[10] H. Chan, M. Luk, and A. Perrig. Using clustering information for sensor network localization. In *The International Conference on Distributed Computing in Sensor Systems (DCOSS 2005)*, 2005.

[11] B. Cheng, R. Hudson, F. Lorenzelli, L. Vandenberghe, and K. Yao. Distributed gauss-newton method for node localization in wireless sensor networks. In *6th Workshop on Signal Processing Advances in Wireless Communications, 2005 IEEE*, pages 915–919, June 2005.

[12] L. Doherty, K. Pister, and L. E. Ghaoui. Convex position estimation in wireless sensor networks. In *NFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655 – 1663, 2001.

[13] A. Doucet, N. D. Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer-Verlag New York, Inc., 2001.

[14] T. Eren, D. Goldenberg, W. Whitley, Y. Yang, A. Morse, B. Anderson, and P. Belheumer. Rigidity, computation, and randomization of network localization. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2673 – 2684, March 2004.

[15] M. Essoloh, C. Richard, and H. Snoussi. Anchor-based distributed localization in wireless sensor networks. In *Workshop on Statistical Signal Processing, 2007. SSP '07. IEEE/SP 14th*, pages 393 – 397, August 2007.

[16] Gasparri A, Pascucci F., and Ulivi G. A distributed extended information filter for Self-Localization in Sensor Networks. In *Proceedings of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, Cannes, France, September 2008.

[17] Gasparri A., Panzieri S., Pascucci F., and Ulivi G.. An interlaced extended Kalman filter for sensor networks localization. In *International Journal of Sensor Networs*, vol. 5(3):164–172, 2009.

[18] Glielmo L., Setola R., and Vasca F. An interlaced extended Kalman filter. In *IEEE Trans. on Automatic Control*, vol. 44(no 8):1546–1549, 1999.

[19] S. Grime, H. Durrant-Whyte, and P. Ho. Communication in decentralized sensing. Technical report, Oxford University Robotics Research Group, 1991.

[20] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95, New York, NY, USA, 2003. ACM Press.

[21] B. Jackson and T. Jordán. Connected rigidity matroids and unique realizations of graphs. *J. Comb. Theory Ser. B*, 94(1):1–29, 2005.

[22] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions ASME Journal of Basic Engineering*, 82:35 – 44, 1960.

[23] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 254–263, New York, NY, USA, 2007. ACM.

[24] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric adhoc routing: of theory and practice. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 63–72, New York, NY, USA, 2003. ACM Press.

[25] V. Lesser, M. Atighetchi, B. Benyo, B. Horling, A. Raja, R. Vincent, T. Wagner, X. Ping, and S. X. Zhang. The intelligent home testbed. *Proceedings of the Autonomy Control Software Workshop (Autonomous Agent Workshop)*, January 1999.

[26] C. Liu, K. Wu, and T. He. Sensor localization with ring overlapping based on comparison of received signal strength indicator. In *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 516 – 518, October 2004.

[27] V. Mehta and M. E. Zarki. A bluetooth based sensor network for civil infrastructure health monitoring. *Wirel. Netw.*, 10(4):401–412, 2004.

[28] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, New York, NY, USA, 2004. ACM Press.

[29] A. G. O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, Inc., Boca Raton, FL, USA, 1998.

[30] D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa. In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE (INFOCOM 2003)*, volume 3, pages 1734 – 1743, April 2003.

[31] Panzieri S., Pascucci F., Setola R., Multirobot Localisation Using Interlaced Extended Kalman Filter In *Proceedings of the 2006 IEEE/RSJ Int. Conf. on Intelligent Robots andSystems (IROS 2006)*, Beijing, China, 2006.

[32] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. H. Growdon, M. Welsh, and P. Bonato. Analysis of feature space for monitoring persons with parkinson's disease with application to a wireless wearable sensor system. In *In Proceedings of the 29th IEEE EMBS Annual International Conference*, Lyon, France, August 2007.

[33] N. Patwari, A. III, M. Perkins, N. Correal, and R. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Transaction on Signal Processing*, 51(8):2137–2148, August 2003.

[34] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Poster abstract: anchor-free distributed localization in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 340–341, New York, NY, USA, 2003. ACM.

[35] C. Savarese, J. M. Rabaey, and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*, pages 317–327, Berkeley, CA, USA, 2002. USENIX Association.

[36] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 201–212, New York, NY, USA, 2003. ACM Press.

[37] X. Shen, Z. Wang, P. Jiang, R. Lin, and Y. Sun. *Advances in Intelligent Computing*, chapter Connectivity and RSSI Based Localization Scheme for Wireless Sensor Networks, pages 578 – 587. Springer Berlin / Heidelberg, 2005.

[38] V. Shnayder, B. rong Chen, K. Lorincz, T. R. F. F. Jones, and M. Welsh. Sensor networks for medical care. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 314–314, New York, NY, USA, 2005. ACM.

[39] M. B. Srivastava, R. R. Muntz, and M. Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving enviroments. In *Mobile Computing and Networking*, pages 132–138, 2001.

[40] I. Stojmenovic. Position-based routing in ad hoc networks. *Communications Magazine, IEEE*, 40(7):128–134, 2002.

[41] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, 2006.

[42] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler. The effects of ranging noise on multihop localization: an empirical study. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 73–80, Piscataway, NJ, USA, 2005. IEEE Press.

[43] Wang Z., Zheng S., Ye Y., and Boyd S. Further Relaxations of the Semidefinite Programming Approach to Sensor Network Localization In *SIAM Journal on Optimization*, vol. 19(2): 655–673, 2008

[44] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan. Ecolocation: A sequence based technique for rf-only localization in wireless sensor networks. In *The Fourth International Conference on Information Processing in Sensor Networks (IPSN '05)*, pages 285 – 292, April 2005.

[45] A. Youssef, A. Ashok, and M. Younis. Accurate anchor-free node localization in wireless sensor networks. In *24th IEEE International Performance, Computing, and Communications Conference, 2005. IPCCC 2005.*, pages 465 – 470, April 2005.

Andrea Gasparri was born in Terni (Italy) in 1979. He graduated cum laude in Computer Science in 2004 from the University of Rome "Roma Tre". Since 2004, he has been with the Department of Computer Science and Automation at the University of Roma "Roma Tre", where he obtained his Ph.D. degree in April 2008 and where he currently holds a postdoctoral position. His current research interests are in the areas of mobile robotics, sensor networks and more generally networked multi-agent systems.

Federica Pascucci was born in Rome (Italy) in 1975. She received the Laurea Degree "cum laude" in 2000 in Computer Science from "Roma Tre" University and the Ph.D. in System Engineering from the University of Rome "La Sapienza" in 2004. She joined the "Roma Tre" University as Assistant Professor in 2005. Her research interests are in the field of mobile robot perception, data fusion and networked robot.