An Interlaced Extended Kalman Filter for sensor networks localisation

A. Gasparri, S. Panzieri*, F. Pascucci and G. Ulivi

Dipartimento di Informatica e Automazione,
Università degli Studi "Roma TRE",
Via della Vasca Navale, 79, Roma, Italy
Email: gasparri@dia.uniroma3.it
Email: panzieri@dia.uniroma3.it
Email: pascucci@dia.uniroma3.it
Email: ulivi@dia.uniroma3.it
*Corresponding author

Abstract: Sensor networks have become a widely used technology for applications ranging from military surveillance to industrial fault detection. So far, the evolution in micro-electronics has made it possible to build networks of inexpensive nodes characterised by modest computation and storage capability as well as limited battery life. In such a context, having an accurate knowledge about nodes position is fundamental to achieve almost any task. Several techniques to deal with the localisation problem have been proposed in literature: most of them rely on a centralised approach, whereas others work in a distributed fashion. However, a number of approaches do require a prior knowledge of particular nodes, i.e. anchors, whereas others can face the problem without relying on this information. In this paper, a new approach based on an Interlaced Extended Kalman Filter (IEKF) is proposed: the algorithm, working in a distributed fashion, provides an accurate estimation of node poses with a reduced computational complexity. Moreover, no prior knowledge for any nodes is required to produce an estimation in a relative coordinate system. Exhaustive experiments, carried on MICAz nodes, are shown to prove the effectiveness of the proposed IEKF.

Keywords: sensor networks; localisation; Kalman filter.

Reference to this paper should be made as follows: Gasparri, A., Panzieri, S., Pascucci, F. and Ulivi, G. (2009) 'An Interlaced Extended Kalman Filter for sensor networks localisation', *Int. J. Sensor Networks*, Vol. 5, No. 3, pp.164–172.

Biographical notes: Andrea Gasparri received the Laurea degree in Computer Science in 2005 and the PhD in Computer Science and Automation in 2008 from the University of "Roma Tre". He is currently working as a Research Assistant in the Robotics Group of the "Dipartimento di Informatica e Automazione" (D.I.A.) at University of "Roma Tre". His research interests are in the field of mobile robotics, sensor networks and multi-agent systems.

Stefano Panzieri received the Laurea degree in Electronic Engineering in 1989 and the PhD in Systems Engineering in 1994, both from the University of Roma "La Sapienza". Since February 1996, he is with the University of "Roma Tre" where currently he is an Associate Professor. His research interests are in the field of industrial control systems and in the area of mobile robots where he studied the problem of localisation in structured and unstructured environments with a special attention to the problem of sensor-based navigation.

Federica Pascucci received the Laurea degree in Computer Science and the PhD in System Engineering at the University of Rome "La Sapienza". In 2005, she joined the University of "Roma Tre" as Assistant Professor. Her research interests are in the field of mobile robot perception, data fusion and networked robot.

Giovanni Ulivi received the Laurea degree in Electrical Engineering in 1974 from the University of Rome "La Sapienza". From that year, he was with Department of Computer and System Science of the same university. In 1992, he moved to the new 3rd University of Rome where he joined the Department of Computer Science and Industrial Automation. His research interests include robotics (in particular autonomous vehicles control) and complex systems. He is member of IEEE and IFAC (International Federation of Automatic Control) Technical Committees on 'Robotics', 'Autonomous Vehicles' and 'Components and Instrument'. He heads the Department.

1 Introduction

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 set of sensors and radio connectivity, 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 fashion. 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.

Some basic services, such as time synchronisation or nodes localisation, must be provided in order to properly set up a sensor network. In fact, basic middle ware services, such as routing, often rely on location information, e.g. geographic routing (Bose et al., 2001; Stojmenovic, 2002; Kuhn et al., 2003). Specifically, the localisation 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 be solved, in fact Jackson and Jordán (2005) and Eren et al. (2004) have proved that a sufficient condition for a sensor network to be localisable cannot be easily identified, even when considering the availability of perfect measurements. Further, several analyses showed that, especially when using the Received Signal Strength Indication (RSSI), having reliable ranging information is fairly practical (Bahl and Padmanabhan, 2000; Whitehouse et al., 2005; Yedavalli et al., 2005).

A taxonomy can be drawn according to the computational organisation of localisation techniques: centralised and distributed. Centralised algorithms exploit a central computer to perform all the complex computations using information gathered by nodes. Distributed algorithms dispense the computation over the network, allowing each node to perform locally and compensating the lack of knowledge through an intensive collaborative processing. Both of them have advantages and drawbacks: centralised algorithms provide good performances but they suffer from structural weakness and do not scale very well, whereas distributed ones often provide sub-optimal solutions but they are also very robust, in terms of resilience, and scalable.

Doherty et al. (2001) propose the Semidefinite Programming Approach (SDP) to solve the localisation problem. The idea is to model geometric constraints between nodes as Linear Matrix Inequalities (LMIs), then use the semidefinite programming theory to solve it. The result is a bounding region for each node, representing feasible locations where nodes are supposed to be. The idea to use a set of convex constraints in order to estimate the position of a node is very elegant, but 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.

Biswas et al. (2006) describe an improved SDP approach to deal with noisy distance measurements. The idea is to take advantage of an additional technique to mitigate the inaccuracy of the solution provided by the SDP formulation. The solution provided by the DSP, 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 approach it is possible to obtain a solution very close to the optimal one. This approach provides a significant improvement in the performance of the SDP-based algorithms. However, the distributed formulation is the result of a clusterisation 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 centralised.

Bulusu et al. (2002) present a RF-based distributed localisation method. The idea is to estimate the location of a node by simply averaging the positions of all the anchors it is connected with. Obviously, the accuracy of the estimation is related to the density of anchors deployed in the environment and the density required to obtain an acceptable estimation is fairly practical.

Moore et al. (2004) 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 localisation 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 in the presence of noise. The algorithm, which does not requires anchors, 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. Moreover, an optional optimisation step can be provided in order to refine the local map first. This algorithm, though performing well, tends to produce orphans because of both the constraints to belong to a quad, and the merging rule.

In this paper, a new approach based on an Interlaced Extended Kalman Filter (IEKF) is proposed. This technique, working in a decentralised fashion, provides an accurate node estimation with an acceptable computational complexity. It does not require any prior knowledge when an estimation on a relative coordinate system is desired. Furthermore, it turns out to be very robust also in presence of noisy distance measurements.

The paper is organised as follows. In section 2, an introduction to the theoretical aspects that have been exploited is provided. In section 3, the proposed IEKF for sensor networks localisation is described. In section 4, the experimental results are proposed. Finally, in section 5, conclusions are presented and future perspectives are discussed.

2 Theoretical background

2.1 Bayesian framework

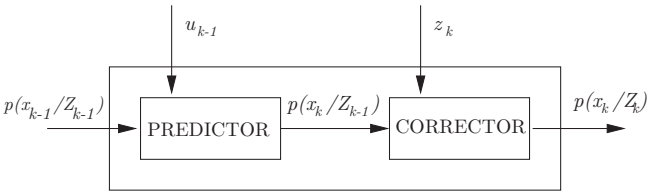
Localisation problem in sensor network can be re-cast into a stochastic estimation problem for a system described by the following equations

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

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 characterise the state transition and the observation z_k respectively.

In a probabilistic form, the localisation problem requires the probability distribution $p(x_k|Z_k,U_k)$ to be computed for all 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. In general, a recursive solution for sensor localisation problem is desirable. Such recursive solution can be achieved applying Bayes filter (see Figure 1).

Figure 1 Bayesian filter



Starting with an estimate for the distribution $p(x_{k-1}|Z_{k-1},U_{k-1})$ at time $k-1$, the joint posterior, following a control u_k and observation z_k, is computed using Bayes theorem. This computation requires that a *state transition model* and an *observation model* are defined, describing the stochastic effects of the control input and observation respectively.

In a probabilistic framework, 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.

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

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

and correction (*measurement update*) form

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

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 localisation algorithm has to resort to additional assumptions.

A common approach is represented by the use of Kalman filter (Kalman, 1960). In this context a linear or linearised system model is required

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

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 modelled 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 discretise the state space. In this way the prediction becomes

$$\begin{aligned} \hat{x}_{k|k-1} &= A_k \hat{x}_{k-1} + B u_k \\ P_{k|k-1} &= A_k P_{k-1|k-1} A_k^T + Q_k \end{aligned} \tag{5}$$

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

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

before update the estimate

$$\begin{aligned} \hat{x}_k &= \hat{x}_{k|k-1} + K_k (z_k - \hat{z}_k) \\ P_k &= P_{k|k-1} - K_k \left[C_k P_{k|k-1} C_k^T + R_k \right] K_k^T \end{aligned} \tag{7}$$

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.

Several probabilistic global methods have been proposed to overcome these drawbacks relaxing Gaussian assumption and introducing the discretisation of the space state. As only Kalman filter is used in the sequel here these techniques are not reported; however, a complete review can be found in Doucet et al. (2001).

2.2 Interlaced Kalman Filter

The Interlaced Kalman Filter (IKF) has been proposed in Glielmo et al. (1999a) to reduce computational load of the estimation process for a class of nonlinear system. The fundamental idea of the IKF is derived from the multi-players dynamic game theory, where the solution of the game is such that each player chooses its strategy as optimal response to the strategy chosen by the other players. IKF is applied to nonlinear system that can be fully linearised by means of an appropriate partition of the state space variables. In this way IKF consist of p parallel KF implementations, each one devoted to estimate only a subset of the state variable, while considering the remaining parts as deterministic time varying parameters. The linearisation error is partially alleviated increasing the noise covariance matrices (Glielmo et al., 1999a)

For sake of clarity, let us consider a system whose system model can be rewritten as (for the first filter $i = 1$ and $j = 2$, while for the second $i = 2$ and $j = 1$)

$$\begin{aligned} x_k^{(i)} &= \tilde{A}_k^{(i)} x_{k-1}^{(i)} + f^{(i)}(x_{k-1}^{(j)}, u_k) + w(i)_k \\ z_k &= C^{(i)}(x_k^{(j)}) x_k^{(i)} + d^{(i)}(x_k^{(j)}) + v_k^{(i)} \end{aligned} \tag{8}$$

where $\tilde{A}_k^{(i)} = A^{(i)} + F^{(ij)}(x_{k-1}^{(j)})$.

The IKF equations proceed from KF filter equations, as is shown in Figure 2. At the *k*-th step, each subfilter form a prediction exploiting both its own estimation and the one of the other filter, according with the following equation

$$\hat{x}_{k|k-1}^{(i)} = \tilde{A}_k^{(i)} \hat{x}_{k-1|k-1}^{(i)} + f^{(i)}(\hat{x}_{k-1|k-1}^{(j)}) \tag{9a}$$

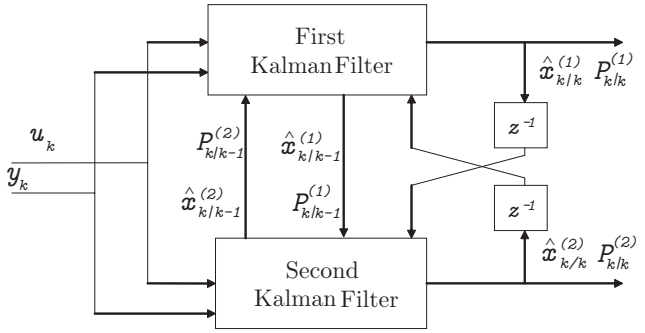
$$P_{k|k-1}^{(i)} = \tilde{A}_k^{(i)} P_{k-1|k-1}^{(i)} \tilde{A}_k^{(i)^T} + Q_k^{(i)} \tag{9b}$$

where

$$\tilde{Q}_k^{(i)} = Q_k^{(i)} + [J_{x,j}^{F,(ij)} + J_{x,j}^{f,(i)}] P_{k-1|k-1}^{(j)} [J_{x,j}^{F,(ij)} + J_{x,j}^{f,(i)}]^T \tag{10}$$

being $J_{x,j}^{F,(ij)}$ and $J_{x,j}^{f,(i)}$ the Jacobians of the relations $F^{(ij)}(x_{k-1|k-1}^{(j)}) x_{k-1|k-1}^{(i)}$ and $f^{(i)}(\cdot)$ with respect to $x_k^{(j)}$.

Figure 2 Interlaced Kalman Filter



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

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_k^{(i)} [z_k - C^{(i)}(\hat{x}_{k|k-1}^{(j)}) \hat{x}_{k|k-1}^{(i)} - d^{(i)}(\hat{x}_{k|k-1}^{(j)})] \tag{11a}$$

$$P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} C^{(i)}(\hat{x}_{k|k-1}^{(j)}) P_{k|k-1}^{(i)} \tag{11b}$$

where the Kalman gain is computed applying the relation

$$K_k^{(i)} = P_{k|k-1}^{(i)} C^{(i)}(\hat{x}_{k|k-1}^{(j)})^T \cdot \tag{12}$$

$$[C^{(i)}(\hat{x}_{k|k-1}^{(j)}) P_{k|k-1}^{(i)} C^{(i)}(\hat{x}_{k|k-1}^{(j)})^T + \tilde{R}_k^{(i)}]^{-1}$$

in which

$$\tilde{R}_k^{(i)} = R_k + \left[J_{x,j}^{C,(i)} + J_{x,j}^{d,(i)} \right] P_{k|k-1}^{(j)} \left[J_{x,j}^{C,(i)} + J_{x,j}^{d,(i)} \right]^T \tag{13}$$

where $J_{x,j}^{C,(i)}$ and $J_{x,j}^{d,(i)}$ are the Jacobians of $C^{(i)}(x_{k|k-1}^{(j)}) x_{k|k-1}^{(i)}$ and $d^{(i)}(\cdot)$ with respect to $x_k^{(j)}$.

From equations (10) and (13) one can notice that the process and measurement noise covariance matrices $Q_k^{(i)}$ and $R_k^{(i)}$ are suitable increased by addition of positive semidefinite quantities that take into account the error introduced by the decoupling operation. As shown in Roumeliotis and Bekey (2002), indeed, it is easy to recognise that the term added to R_k in equation (13) represents the cross-correlation between the filters due to innovation process, while the term added to $Q_k^{(i)}$ in (10) is related to the cross-correlation induced by propagation process.

Notice that, in a deterministic framework, sufficient conditions that guarantee the local convergence of the estimator are established in Glielmo et al. (1999b).

This formulation of IKF assumes that both substate transition mapping and observation mapping, i.e. equations (8), depend affinely on their arguments. If one removes these assumptions, the algorithm can be still applied by linearising, at each step, every subsystem obtaining the IEKF.

3 IEKF for sensor network

In this work, a group of *N* nodes is deployed on a planar environment. The nodes, that are supposed to be static, are equipped with rangefinder sensors and limited-coverage wireless devices in order to share information each others.

Specifically, if a node *i* is in the coverage area of the node *j*, they are able to exchange data, i.e. their estimated positions and the related uncertainty. In this way node *i* acquires information on its relative position with respect to node *j*, and *vice versa.*

Few nodes are equipped with absolute positioning devices and there is no need to compute their positions, but they play the role of anchors in the network. Therefore their locations are assumed to be *a priori* known.

Localisation problem is regarded in the framework of stochastic estimation. The state to be estimated is represented by the positions of the nodes

$$\mathbf{x}_k \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \\ \vdots \\ x_k^{(N)} \end{bmatrix} \tag{14}$$

where $x_k^{(i)} = [p_{x,k}^{(i)}, p_{y,k}^{(i)}]^T$ is the position of the *i*-th node in a global reference frame. The measurements of the system are the relative distances retrieved by rangefinders. All uncertainty sources are assumed to have Gaussian distribution.

An IEKF is applied to estimate the location of the nodes. In particular *N* parallel EKFs are implemented. Each one runs on a node and is devoted to estimate its locations.

3.1 Prediction model

As mentioned above, the node are assumed to be static

$$x_k^{(i)} = x_{k-1}^{(i)} + w_k^{(i)} \tag{15}$$

where $w_k^{(i)} \in \mathcal{R}^2$ is a zero mean white noise vector with covariance matrix $Q_k^{(i)}$.

As the state transition of each node does not affect the location of the other nodes, in (9b) the term $\tilde{Q}_k^{(i)}$ is not computed.

3.2 Observation model

The node observations consist in the rangefinder measurements. This last is composed by n_i sub-vectors

$$z_k^{(i,j)} = h^n(x_k^{(i)}, x_k^{(j)}) = \sqrt{(p_{x,k}^{(j)} - p_{x,k}^{(i)})^2 + (p_{y,k}^{(j)} - p_{y,k}^{(i)})^2} \quad (16)$$

where n_i is the number of nodes in the viewing area of the i-th node and n_i^a sub-vectors

$$z_k^{(i,j)} = h^a(x_k^{(i)}, \mathcal{M}) = \sqrt{(l_x^{(j)} - p_{x,k}^{(i)})^2 + (l_y^{(j)} - p_{y,k}^{(i)})^2} \quad (17)$$

where n_i^a is the number of anchors in the viewing area of the i-th node and $(l_x^{(j)}, l_y^{(j)})$ the position of an anchor. It should be noted that $h^a(\cdot)$ depends on the way in which environment map is represented; in our case a list of anchors position $(l_x^{(j)}, l_y^{(j)})$. .

Due to the nonlinearity of the mapping, the Jacobian of the maps $h^n(\cdot)$ and $h^a(\cdot)$ have to be used, instead of matrix $C^{(i)}(\cdot)$, in equations (11) and (13).

When a node detects another one, the covariance update is calculated according to equation (11b). As the position of an anchor does not affect the location of the node, in equation (13) the term $\tilde{R}_k^{(i)}$ is not computed, when the measurement detect an anchor.

3.3 Complexity analysis

It is well known that the main drawbacks related with implementation of localisation algorithms based on EKF approaches are due to huge computational load and memory occupancy. Indeed, both these quantities scales as $\sim O(N^2)$ being N the number of the nodes.

The formulation proposed above seems to be more efficient. Memory occupancy scales linearly with the number of nodes, i.e. $\sim O(N)$. The computational load is distributed on the nodes. For each node i, it depends on the number of the nodes n_i and the anchors n_i^a in the viewing area, and scales linearly on this, then as $\sim O(n_i + n_i^a)$.

This is an interesting feature, as, even if the solution obtained by means of this algorithm is sub-optimal, it represents a trade-off between estimation accuracy and computational requirements suitable with the limited hardware resources of nodes.

4 Performance analysis

Several simulations have been executed in order to investigate the underlying properties of the proposed IEKF, such as the accuracy and the robustness of the estimation. In order to achieve that, a simulation software able to generate suitable test cases has been exploited. Particular attention has been devoted to susceptibility of the algorithm to environmental factors. Specifically, the following aspects have been taken into account:

- density of anchor deployment
- density of node deployment
- level of noise of observations

Figure 3 shows the result when considering a variable number of anchors, ranging from 1 to 9, with a fixed number of nodes 70. According to this result, the algorithm performs better, in terms of estimation accuracy and convergence rate, when considering an increasing number of anchors. In detail, two different behaviours can be recognised, considering anchors ranging from 1 to 3 or from 5 to 9. This allows to define an optimal number of anchors to be used for a real deployment with respect to some parameters of interest.

Figure 3 Estimation error vs. density of anchor deployment: variable number of anchors (from 1 to 9), fixed number of nodes (70) (see online version for colours)

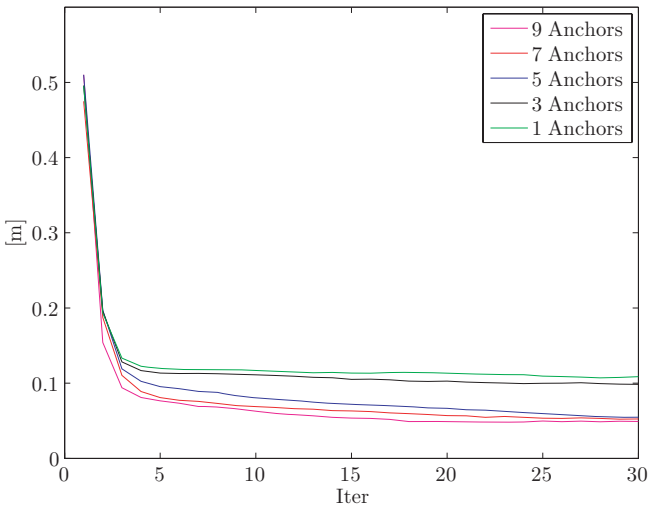


Figure 4 shows the result when considering a variable number of nodes, ranging from 10 to 90, with a fixed number of anchors 5. According to this result, the algorithm performs slightly better, in terms of convergence rate, when considering an increasing number of nodes. However, no significant improvement can be noticed in relation of the accuracy of the estimation. It can be justified considering that the accuracy is mainly related to the number of available anchors and the noise of observations.

Figure 4 Estimation error vs. density of nodes deployment: variable number of nodes (from 10 to 90), fixed number of anchors (5) (see online version for colours)

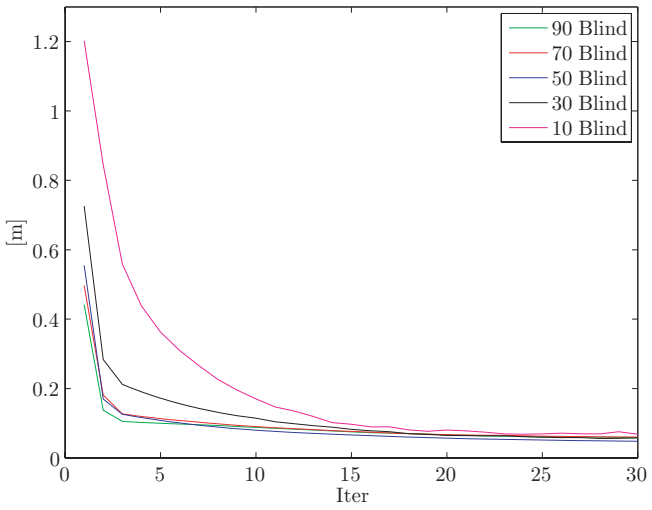
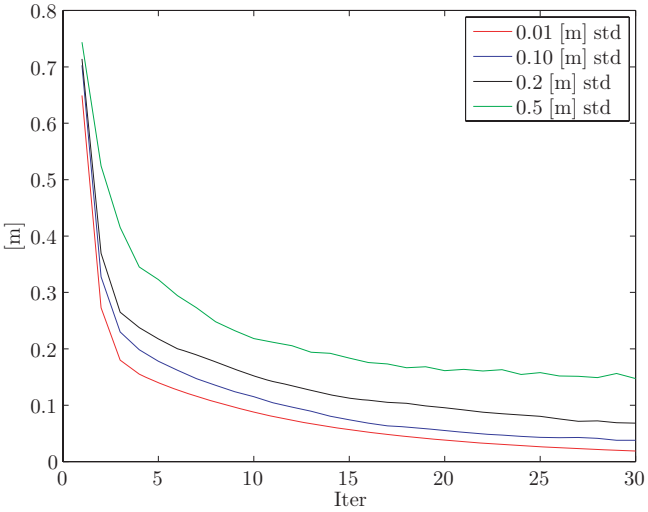


Figure 5 shows the result when considering a variable level of noise with a std ranging from 0.01 m to 0.5 m, with both a fixed number of anchors 5 and nodes 30. According to this result, the algorithm performs better, in terms of convergence rate, when considering a decreasing level of noise.

Figure 5 Estimation error vs. level of noise of observations: variable level of noise (std ranging from 0.01m to 0.5m), fixed number of anchors (5), fixed number of anchors (30) (see online version for colours)



5 Experimental results

In order to prove the effectiveness of the proposed decentralised IEKF in a real context, several experiments have been carried out in an indoor environment. The network has been built with the MICAz (MPR2400) platform, a generation of Motes from Crossbow Technology. The MPR2400 (2400 MHz to 2483.5 MHz band) uses the Chip-con 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.

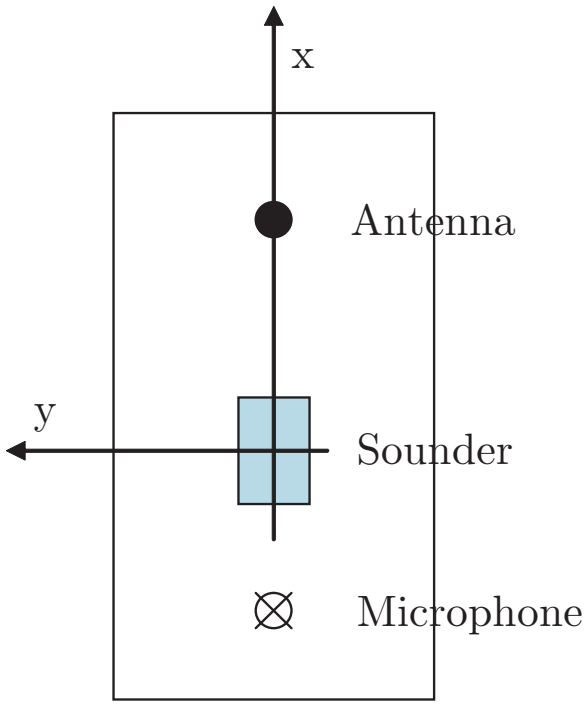
MICAz platform comes along with TinyOS, an open-source event-driven operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimising code size as required by the severe memory constraints inherent 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.

5.1 Ranging technique

The mechanism adopted to measure the inter-node distance is the Time Difference of Arrival (TDoA). Having each node equipped with a speaker and a microphone, TDoA mechanisms compute the distance among nodes by determining the difference between the time of arrival of two pulses, characterised by a different propagation velocity.

As previously mentioned, several sensors and acquiring boards can be connected to the MICAz (Figure 6) platform. In particular, two different boards – the MTS300 and the MTS310 – provide a sounder as well as a microphone. 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. As a consequence, according to this hardware availability, RF and acoustic (sounder) signals have been exploited.

Figure 6 Configuration of MICAz device (see online version for colours)



The proposed ranging technique has been thoroughly investigated to understand its accuracy and consequently to provide a better evaluation of the algorithm capabilities. A significant amount of inter-node distances, considering also different environmental conditions, were gathered and statistical analyses were performed. In regards to the experiments in the lab bench, the ranging technique can achieve a precision of around 3 ~ 8 cm with a standard deviation of 8 ~ 14 cm according to the measured distance (ranging from 20 cm to 2.5 m).

Moreover, experiments have been performed to verify if the mutual orientation of nodes can influence the distance measured. For such a reason, two nodes were arranged on the floor at the distance of 54 cm from each other. Specifically, the distance was manually measured from the sounder of the emitter to the microphone of the receiver. Afterward, data was collected considering different orientations of nodes, in order to simulate a realistic random deployment on the ground.

Table 1 shows the statistic results using more than 200 measurements for each configuration. According to the experimental results, there are no significant variations on the obtained measures when considering different mutual orientations. However, as mentioned above, data presents a bias as well as a considerable standard deviation that makes their use challenging.

Table 1 Inter-node ranging technique: experimental results

Exp.	*Mean value*	*SD*	*Node 1 orientation*	*Node 2 orientation*
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

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 characterise 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.

5.2 Evaluation criteria

In order to evaluate the effectiveness of the proposed algorithm, two indexes of quality have been considered for each node and for each axes: the estimation error, computed using the Euclidean distance as a metric and the estimation covariance. Moreover, some global indexes have been also taken into account: maximum, minimum and average error of estimation, the velocity of convergence and finally, the percentage of estimation failures. The first two indexes give an idea about the local algorithm behaviour, whereas the other ones give an evaluation of the global algorithm performance.

In order to have a better evaluation of the proposed IEKF, a comparison against two other approaches has been carried out. To make the comparison fair, algorithms were executed batch under the same conditions exploiting Matlab code developed by authors. In particular, 100 trials were run for each configuration. Afterwards, the collected data were used to compute the indexes of interest previously described.

The first comparison was against a centralised version of an Extended Kalman Filter with the aim of better understand the advantages as well as the drawbacks that arise when decentralising an algorithm.

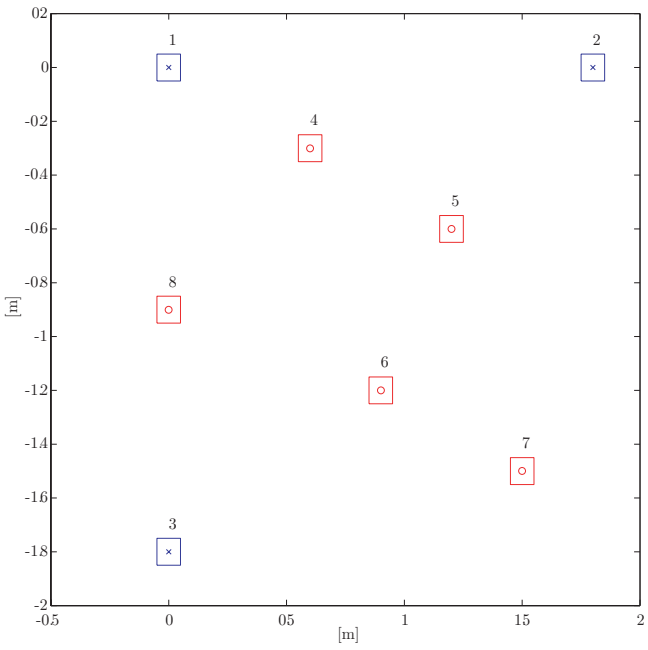
The second comparison was against the algorithm for Relative Location Estimation proposed in Patwari et al. (2003). In detail, it is a Maximum-Likelihood Estimator (MLE) that works in a centralised fashion. The Matlab code, that is freely available, can be downloaded at http://www.eecs.umich.edu/~hero/localize/.

5.3 Network deployment

The robotics laboratory of University of "Roma Tre" has been exploited for the network deployment. Two different configurations have been built and several data acquisitions have been done. Moreover, different anchors locations have been considered in order to understand the performances when changing the configuration. Real locations were measured manually taking advantage of the regularity of the flooring grid.

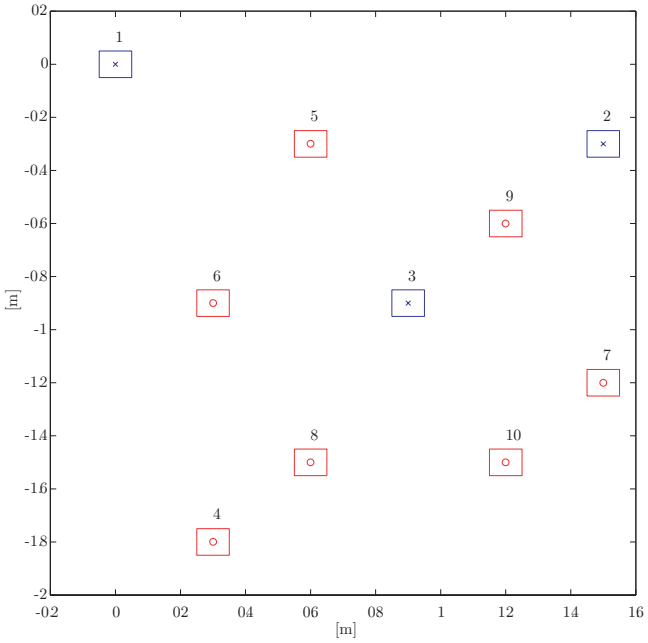
Specifically, Figure 7 describes the first configuration that has been considered. Here, three anchors were deployed on the border of the network so that optimal nodes coverage could be guaranteed. In addition, each node was ideally within the communication range of each other so that a full connected graph was available.

Figure 7 First configuration for the network deployment: anchors (1–3) were arranged on the border, while nodes to be localised (4–8) were randomly positioned (see online version for colours)



Further, Figure 8 describes the second configuration that has been exploited. In this case, anchors (always three) were deployed so that collinear arrangements could be found and an optimal coverage of the network could not be guaranteed. Again, each node was ideally within the range of communication of each other in order to have a full connected graph.

Figure 8 Second configuration for the network deployment: anchors (1–3) as well as nodes to be localised (4–10) were almost randomly arranged (see online version for colours)



5.4 *IEKF vs. EKF*

Here a comparison of the proposed Interlaced Extended Kalman Filter (EKF) against a centralised Extended Kalman Filter (IEKF) is provided.

In regards to the first configuration shown in Figure 7, the results of the centralised and distributed algorithms averaged over 100 trials are collected in Table 2. Here a synoptic comparison between the two approaches can be found. According to the accuracy of available data, both algorithms are able to localise all nodes within the network with similar performances.

Table 2 Statistical analysis for the first configuration: indexes of quality

Conf. 1	*EKF*	*IEKF*
Max Error [m]	0.1104	0.1490
Min Error [m]	0.0132	0.0172
Mean Error [m]	0.0619	0.0715
Converg. Rate	39	28
% Failure	0	0

The similarity of performances can be related to the good coverage of the network provided by this anchors deployment. In fact, such a deployment is able to make up the lack of knowledge for the decentralised algorithm with a more significant set of data.

As one can expected, EKF performs slightly better, especially in terms of minimum, maximum and average error.

This behaviour can be easily explained with the different amount of data available for the two algorithms as well as with the more complete interpretation of data typical of a centralised approach, that takes advantage from the complete knowledge of cross-correlation terms. The maintenance of these terms increases the convergence rate of the EKF that results slower than the one of the IEKF.

From a complexity point of view, as explained in Section 3.3, the IEKF presents a reduced memory occupancy. Due to the small number of nodes involved in the localisation process, the differences on the computational load cannot be appreciated, as a full update of centralised algorithm takes more or less the same time of a full update of IEKF, as reported in Table 3. It should be noticed, however, that the IEKF update can be split in n_i smaller updates running independently on different processors, while the same parallelism cannot be achieved by EKF.

Table 3 Full update time over an Intel® Pentium M 725 (1.6 GHz)

Conf.	*EKF [s]*	*IEKF [s]*
1	0.0024	0.0021
2	0.0048	0.0027

The results obtained using the second configuration are summarised in Table 4.

Table 4 Statistical analysis for the second configuration: indexes of quality

Conf. 2	*EKF*	*IEKF*
Max Error [m]	0.1522	0.3153
Min Error [m]	0.0301	0.0314
Mean Error [m]	0.0786	0.1176
Converg. Rate	35	22
% Failure	1	2

Table 5 Statistical analysis for the first configuration: indexes of quality

Conf. 1	*MLE*	*IEKF*
Max Error [m]	0.1171	0.1490
Min Error [m]	0.0540	0.0172
Mean Error [m]	0.0805	0.0715

Table 6 Statistical analysis for the second configuration: indexes of quality

Conf. 2	*MLE*	*IEKF*
Max Error [m]	0.2256	0.3153
Min Error [m]	0.0568	0.0314
Mean Error [m]	0.1041	0.1176

This table collects the indexes of qualities for this experiment. In this case, in which anchors were deployed so that collinear arrangements were possible and a complete coverage of the network were not guaranteed, some differences between the two algorithms can be noticed. In particular, while the centralised algorithm is able to localise all nodes, the decentralised one does not provide a good estimation for node 4. This can be explained with the fact that node 4 is on the border of the network. As a consequence, having data

coming only from one side, its estimation is more susceptible to biases. Note also that such anchors deployment implies a limited percentage of failures.

From a computational point of view, it can be notice from Table 3 that the execution time of the EKF update grows exponentially, as the number of nodes increases.

5.5 IEKF vs. MLE

Here a comparison of the proposed Interlaced Extended Kalman Filter against the centralised Maximum-Likelihood Estimator (MLE) proposed in Patwari et al. (2003) is provided. Let us note that, in order to use the software provided by the authors of Patwari et al. (2003), the real data collected by Motes needed to be adjusted. In detail, all missing observations have been replaced with a reliable estimation of the real distance as the provided code did not take into account failure in the observing process.

6 Conclusions

In this paper a decentralised IEKF to solve the localisation problem in sensor networks has been proposed. The algorithm, working in a decentralised fashion, provides an accurate estimation with an acceptable computational complexity. Several experiments have been executed in order to prove its effectiveness. A comparison with a centralised version of the Extended Kalman Filter has been provided and typical behaviours for both algorithms, centralised and decentralised, have been shown. Furthermore, a statistical analysis over 100 trials for each configuration has been performed to validate the robustness of the decentralised IEKF.

According to the experimental results, the centralised EKF performs, in the average, slightly better. This behaviour can be related to the more complete interpretation of data given by a centralised approach. However, the decentralised IEKF gives similar results with a significant reduction of the computational complexity. Therefore, the IEKF, working in a decentralised fashion, turns out to be a robust as well as a flexible framework, suitable to the collaborative processing paradigm typical of the sensor network philosophy.

Several interesting challenges still remain for future works. First of all, an analysis to evaluate the tracking capability of the proposed algorithm, when a target is moving within the network, will be faced. Successively, these results will be integrated to face the problem of closing the control loop for a robot moving within the networks.

References

Bahl, P. and Padmanabhan, V.N. (2000) 'Radar: An in-building RF-based user location and tracking system', *INFOCOM*, Vol. 2, pp.775–784.

Biswas, P., Liang, T-C., Toh, K-C., Ye, Y. and Wang, T-C. (2006) 'Semidefinite programming approaches for sensor network localization with noisy distance measurements', *IEEE Transaction on Automation Science and Engineering*, Vol. 3.

Bose, P., Morin, P., Stojmenovic, I. and Urrutia, J. (2001) 'Routing with guaranteed delivery in ad hoc wireless networks', *Wireless Networks*, Vol. 7, No. 6, pp.609–616.

Bulusu, N., Bychkovskiy, V., Estrin, D. and Heidemann, J. (2002) 'Scalable, ad hoc deployable RF-based localization', *Grace Hopper Celebration of Women in Computing Conference 2002*, Vancouver, British Columbia, Canada, University of California at Los Angeles.

Doherty, L., Ghaoui, L.E. and Pister, K.S.J. (2001) 'Convex position estimation in wireless sensor networks', *Proceedings of IEEE Infocom.*

Doucet, A., de Freitas, J. and Gordon, N. (2001) *Sequential Montecarlo Methods in Practice*, SpringerVerlag, NewYork.

Eren, T., Goldenberg, D., Whitley, W., Yang, Y., Morse, A., Anderson, B. and Belheumer, P. (2004) 'Rigidity computation and randomization of network localization', *Proceedings of IEEE Conference on Computer Communication* (*Infocom 2004*).

Glielmo, L., Setola, R. and Vasca, F. (1999a) 'An interlaced extended kalman filter', *IEEE Transactions on Automatic Control*, Vol. 44, No. 8, pp.1546–1549.

Glielmo, L., Setola, R. and Vasca, F. (1999b) 'Interlaced extended kalman filter as deterministic nonlinear observer', *Proceedings of European Control Conference ECC'99*, Karlsruhe, Germany.

Jackson, B. and Jordán, T. (2005) 'Connected rigidity matroids and unique realizations of graphs', *Journal of Combinatorial Theory Ser. B*, Vol. 94, No. 1, pp.1–29.

Kalman, R. (1960) 'A new approach to linear filtering and prediction problems', *Transactions ASME Journal of Basic Engineering*, Vol. 82, pp.35–44.

Kuhn, F., Wattenhofer, R., Zhang, Y. and Zollinger, A. (2003) 'Geometric ad-hoc routing: of theory and practice', *Twenty-second ACM Symposium on Principles of Distributed Computing (PODC).*

Moore, D., Leonard, J., Rus, D. and Teller, S. (2004) 'Robust distributed network localization with noisy range measurements', *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*', ACM Press, New York, pp.50–61.

Patwari, N., III, A.O.H., Correal, M.P.N. and O'Dea, R.J. (2003) 'Relative location estimation in wireless sensor networks', *IEEE Transactions on Signal Processing*, Vol. 51, No. 8, pp.2137–2148.

Roumeliotis, S. and Bekey, G. (2002) 'Distribuited multirobot localization', *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, pp.781–795.

Stojmenovic, I. (2002) 'Position-based routing in ad hoc networks', *IEEE Communication Magazine*, Vol. 40, pp.128–134.

Whitehouse, K., Karlof, C., Woo, A., Jiang, F. and Culler, D. (2005) 'The effects of ranging noise on multihop localization: an empirical study', *International Symposium on Information Processing in Sensor Networks*, pp.73–80.

Yedavalli, K., Krishnamachari, B., Ravula, S. and Srinivasan, B. (2005) 'Ecolocation: a sequence based technique for RF localization in wireless sensor networks', *The 4th International Conference on Information Processing in Sensor Networks* (*IPSN 2005*), Los Angeles, CA.