A new Extended Kalman Filter based on actual local information for mobile robots

Giuseppe Cotugno, Luigi D'Alfonso, Pietro Muraca, Paolo Pugliese

DEIS, Università della Calabria, 87030 Rende (Cs), Italy
(cot.giu@gmail.com, {ldalfonso, muraca, pugliese}@deis.unical.it)

Abstract: In this work we propose a new version of Extended Kalman Filter for robot localization. Such algorithm is based on local data and does not require any assumption on robot's working environment. The filter is driven by measurements, taken by ultrasonic sensors located onboard the robot, and a switching sensors activation policy is devised, which allows power saving still achieving accurate tracking. We obtain reliable estimation results, as shown in simulations. The proposed algorithm works well both in "simple" and in "complex" environments.

Keywords: Mobile robots, State estimation, Extended Kalman filters, Sensor fusion.

1. INTRODUCTION

Robot navigation is a well known problem that involves many aspects of different topics. In order to perform a reliable navigation, the robot needs

(1) to build a map of the environment
(2) to localize itself within such map

The two problems are related to each other. In order to know where the robot is, a good environment mapping is needed; in order to have an effective environment mapping, the robot's position and orientation should be estimated as best as possible because mapping is performed from measures and information relative to such estimated position and orientation.

Many solutions have been proposed in literature for solving both problems simultaneously [11]. Specifically, many of those can be classified as solutions to the SLAM (Simultaneous Localization And Mapping) problem. Considering the second problem, there are many works in literature dealing with mobile robot localization in a totally or partially known environment. In most of these works mobile robots are equipped with a set of ultrasonic sensors, placed onboard; these sensors measure (within tolerance) the distance of the robot from the boundaries of the environment where they move. Consequently, whenever robot sensors are placed onboard, an environment model is required in order to estimate robot position and orientation.

A well known technique used for robot localization is the Extended Kalman Filter. Such filter helps estimate robot (position and orientation) state given a model for robot evolution and sensors measurements. Considering the last feature, however, if the environment is unknown, providing a measurements' model is more difficult.

In contrast of [2], in this work we assume that the robot working environment is totally unknown and we use local information to approximate a good measurements' model to support the EKF.

The basic idea behind our algorithm is that is not needed to know the whole environment surrounding the robot to estimate its current position and orientation; to obtain the estimate only portions of the environment that interact with robot's sensors are needed. This concept has been implemented as *Neighbours Based Algorithm* and will be illustrated in section 3.

Another issue that we consider is a sensor switching policy. Most of mobile robots are equipped with a large number of sensors. All these sensors provide measurements that can be fused to obtain a reliable robot state estimate. Although it seems intuitive that the more sensors you use, the better estimate you get, there are reasons that suggest not to exceed with their number or use.

The most important reason is that sensors employ robot batteries, and an intensive use reduces the autonomy of the robot. If the robot uses ultrasonic sensors, there are also situations where multiple sonar sensors cannot operate simultaneously, for example when they use the same frequency band [1].

When the filter is not run onboard the robot, which is the most frequent case, one more issue is bandwidth consumption and possible collisions when transferring measurements from the sensors to the filter.

These considerations impose to find a trade-off between the number of sensors that are active at each instant and the accuracy of the state estimate. In other words we are focused on finding the "best" sequence of activation of a small (fixed) fraction of the available sensors.

The measurements taken by the sensors will be used by a Kalman filter to provide the state estimate; thus an activation sequence will be "better" than another if it will give an estimate of the state that is more accurate and offers better statistical properties.

The paper is organized as follows: in section 2 the model of the mobile robot is presented; in section 3 we present our new version of the EKF and in section 4 we propose a

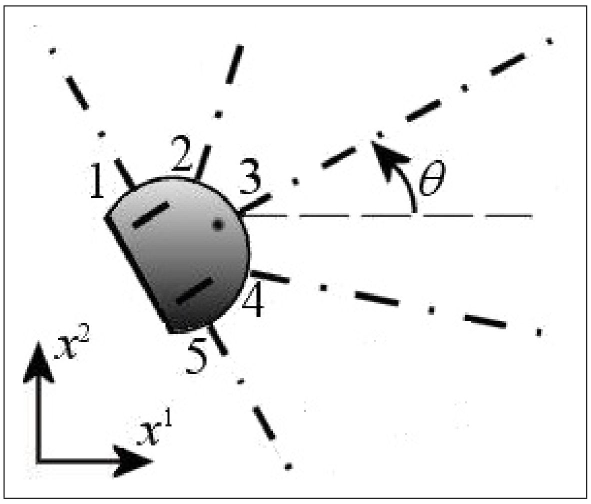

Fig. 1. Robot sketch.

switching technique to improve the accuracy of the state estimate. Finally in section 5 and 6 we present and discuss the results of the numerical simulations and outline some possible extensions for future investigations.

2. ROBOT MODEL

We consider a mobile robot with two independent driving wheels and one castor wheel, as shown in Figure 1: a concrete example is the Khepera III robot [2] used in many academic laboratories. For such a robot, an approximated, discrete-time model, which neglects the dynamic of the motors and the frictions, is [3]:

$$\begin{cases} x^1_{k+1} = x^1_k + v_k T\cos(\theta_{k+1}) + w^x_k \\ x^2_{k+1} = x^2_k + v_k T\sin(\theta_{k+1}) + w^y_k \\ \theta_{k+1} = \theta_k + \Delta_k + w^\theta_k, \end{cases} \tag{1}$$

where:

- (x^1_k, x^2_k) is the position of the robot at time t_k
- θ_k is the angle between the robot axle and the x-axis
- $v_k = R\,(\omega^l_k + \omega^r_k)/2$ is the linear velocity of the robot
- ω^l_k and ω^r_k are the angular velocities of wheels
- w^x_k, w^y_k, w^θ_k are zero-mean uncorrelated Gaussian noises
- R is the radius of the wheels
- l is the length of the axes
- $\Delta_k = R\,(\omega^l_k - \omega^r_k)T/l$ is the rotation within $[t_{k-1}, t_k]$
- $T = t_k - t_{k-1}$ is the sampling period.

The input variables of the model are the angular velocities of the wheels ω^l_k and ω^r_k, and they have been precomputed to make the robot follow the desired trajectories. The model's state variables are $x_k = [x^1_k \quad x^2_k \quad \theta_k]'$. The Gaussian disturbances take into account unmodeled dynamics, friction, wheels slipping and also, if the case, external disturbances such as wind.

Observation structure

We suppose the robot is equipped with five ultrasonic sensors (this is the case of the Khepera III), located as shown in Figure 1 and denoted by S_i, $i = 1, \ldots, 5$. In contrast of [2], we assume the robot placed in an unknown

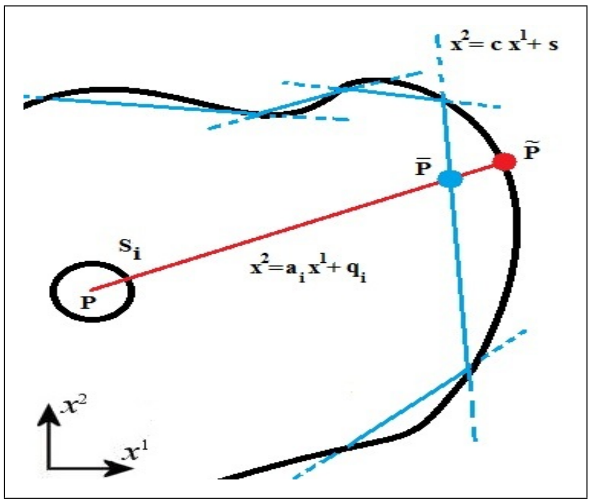

Fig. 2. Observation model.

environment, therefore we haven't any model for the observation structure. We model the workspace as a set of segments such that each of them intersects at least at one point of the boundary of the environment (see Figure 2). Each sensor provides the distance between the center of gravity of the robot, denoted by $P = (x^1_k, x^2_k)$, and one point on the environment, denoted by $\tilde{P} = (\tilde{x}^1, \tilde{x}^2)$.
Considering the model used for the environment, the measure provided by the i-th sensor, S_i, is approximated, as shown in Figure 2, by the distance between P and the intersection, denoted by $\overline{P} = (\overline{x}^1, \overline{x}^2)$, between the axis of the sensor S_i and one of the representative segments of the world.
Marking the axis of the sensor S_i as $x^2 = a_i x^1 + q_i$ and denoting the axis of the representative segment as $x^2 = cx^1 + s$, the intersection $\overline{P}$ is given by

$$\overline{x}^1 = \frac{s - q_i}{a_i - c} \qquad \overline{x}^2 = \frac{a_i s - c q_i}{a_i - c}\,, \tag{2}$$

therefore the measure, y_i, provided by sensor S_i, is modeled as

$$y_i = \sqrt{(x^1_k - \tilde{x}^1)^2 + (x^2_k - \tilde{x}^2)^2}\,, \tag{3}$$

and approximated by

$$y_i \approx \sqrt{(x^1_k - \overline{x}^1)^2 + (x^2_k - \overline{x}^2)^2}\,. \tag{4}$$

We perfectly know the robot structure, therefore we know the orientation, α_i, of each sensor S_i, with respect to robot axis (placed on third sensor axis), so the axis of the sensor S_i is given by:

$$\begin{aligned} a_i &= \tan(\theta_k + \alpha_i) \\ q_i &= x^1_k - a_i x^2_k\,. \end{aligned} \tag{5}$$

By replacing (4) with (5), we obtain an observation function, y_i, depending only on robot state and segment (s, c), such that

$$y_i = h((x^1_k, x^2_k, \theta_k), (s, c)), \quad i = 1, \ldots, 5$$

These relationships define the output equation of the dynamical model of the robot and will be written in the more compact form

$$y_k = h(x_k, (\overline{s}, \overline{c})) + v_k, \tag{6}$$

where the dimension of vector y_k ranges from one to five, depending on how many sensors we use, the vector v_k collects the sensor noises, also assumed Gaussian and zero-mean, and uncorrelated with w_k. The vectors $\overline{s}$ and $\overline{c}$ contain representative segments' parameters (s, c), one segment for each sensor used.

3. NEIGHBOURS BASED EXTENDED KALMAN FILTER (NEKF)

Nonlinear filtering is the problem of estimating the state of a nonlinear stochastic system from noisy measurements. For discrete-time systems such framework is represented by the equations

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k \\ y_k &= h(x_k) + v_k, \end{aligned} \tag{7}$$

where x is the state to be estimated, y is the measured output, w and v are the system and measurements noises. Ignoring (s, c), model (1), (6) falls within this framework on defining

$$x_k = [x_k^1 \;\; x_k^2 \;\; \theta_k]' \quad w_k = [w_k^x \;\; w_k^y \;\; w_k^\theta]' \quad u_k = [\omega_k^l \;\; \omega_k^r]'.$$

In the sequel we will denote by W and V the covariance matrices of the noises w_k and v_k, respectively. These matrices will be assumed known. Moreover we suppose we are given an initial estimate of the state $\hat{x}_{0|0}$ and a measure of its "reliability" by its estimation error covariance matrix $P_{0|0}$.

The above observation structure (6), has a meaning only if (s, c) are known. In order to approximate (s, c), we designed an algorithm: the Neighbours Based Algorithm (NBA).

3.1 Neighbours Based Algorithm

The core idea behind the NBA is proximity among acquired measures. Given a new measure y_i, acquired by sensor S_i, this measure is connected to one point of the environment, P_i^*, such that $||P - P_i^*|| \approx y_i$, where P_i^* is an approximation, due to the presence of noise v_k, of $\tilde{P}$ defined in the Observation Structure section.

Starting with P_i^*, the NBA searches a subset of points among all previous acquired environment's points. The points found will be *close to (neighbours of)* P_i^*.

We define two points P_1, P_2 as neighbours iff

$$||P_1 - P_2|| < r,$$

where r is a given algorithm parameter.

Therefore we defined the closeness function $f_{\mathcal{N}}$ as:

Given a set of points $\mathcal{A}$ and a point P,

$$f_{\mathcal{N}}(P, \mathcal{A}) = \mathcal{B}$$

where $\mathcal{B} \subseteq \mathcal{A}$ and $\mathcal{B} = \{P_i \in \mathcal{A} : ||P_i - P|| < r\}$.

Moreover we define the $LMS(\cdot)$ function as follows:

Given a set of points $\mathcal{B}$

$$LMS(\mathcal{B}) = (m, q)$$

where (m, q) are the Least Mean Squared approximation of the straight line through points in $\mathcal{B}$

After defining the closeness function $f_{\mathcal{N}}$ and the $LMS(\cdot)$ function, we summarize the NBA algorithm as follows:

Neighbours Based Algorithm

```
At each step, given $\hat{x}_k^1, \hat{x}_k^2, \hat{\theta}_k, \mathcal{M}_{k-1}, I_k$, do
(1) for $i \in \mathcal{I}_k$
        • acquire measure $y_k^i$ from sensor $S_i$
        • $P_{i,1}^* = \hat{x}_k^1 + y_k^i \cos\hat{\theta}_k$
        • $P_{i,2}^* = \hat{x}_k^2 + y_k^i \sin\hat{\theta}_k$
        • $P_i^* = (P_{i,1}^*, P_{i,2}^*)$
    end
(2) $\mathcal{M}_k = \mathcal{M}_{k-1} \cup \{\cup_{i \in \mathcal{I}_k}\{P_i^*\}\}$
(3) for $i \in \mathcal{I}_k$
        • $\mathcal{N}(P_i^*) = f_{\mathcal{N}}(P_i^*, \mathcal{M}_k)$
        • $(\hat{s}_i, \hat{c}_i) = LMS(\mathcal{N}(P_i^*))$
    end
```

where

- $\mathcal{I}_k$ is the set of sensor indexes to be used at step k
- $\hat{x}_k^1, \hat{x}_k^2, \hat{\theta}_k$ is the robot state estimation at step k
- $\mathcal{M}_{k-1}$ is the set of previously acquired environment points
- $(\hat{s}_i, \hat{c}_i)$ are the approximations, at step k, of the environment representative segment parameters

From now on we will use the NBA algorithm as a function:

$$(\hat{s}_i, \hat{c}_i) = NBA(i, k),$$

where i is the sensor index and k is the time step.

3.2 Neighbours based Extended Kalman Filter

The Extended Kalman Filter (EKF) has been used for many years to estimate the state of a nonlinear system from noisy measurements, and it has been probably the first concrete application of Kalman work on filtering [5]. It is based on the linearization of the nonlinear maps (f, h) of (7) around the estimated trajectory, and on the assumption that the initial state and measurement noises are Gaussian and uncorrelated each other.

From the computational point of view the EKF is simply a time-varying Kalman filter where the dynamic and output matrices are given by

$$A_k = \left.\frac{\partial f(x, u_k)}{\partial x}\right|_{x=\hat{x}_{k|k}}, \quad C_k = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}_{k|k-1}}, \tag{8}$$

and its output is a sequence of state estimates $\hat{x}_{k|k}$ and matrices $P_{k|k}$; starting from given $(\hat{x}_{0|0}, P_{0|0})$ its steps are [4]:

Neighbours based Extended Kalman Filter

$$\begin{aligned}
&\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_k) \\
&P_{k+1|k} = A_k P_{k|k} A_k' + W \\
&K_{k+1} = P_{k+1|k} C_{k+1}' (C_{k+1} P_{k+1|k} C_{k+1}' + V)^{-1} \\
&(\hat{\overline{s}}_k, \hat{\overline{c}}_k) = NBA(\mathcal{I}_k, k) \\
&\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(y_{k+1} - h(\hat{x}_{k+1|k}, (\hat{\overline{s}}_k, \hat{\overline{c}}_k))) \\
&P_{k+1|k+1} = P_{k+1|k} - K_{k+1} C_{k+1} P_{k+1|k}
\end{aligned}$$

There $\hat{x}_{k+1|k}$ represents the estimate of x_{k+1} *before* getting the observation y_{k+1}, $\hat{x}_{k+1|k+1}$ represents the estimate *after* getting that observation, and $(\hat{\bar{s}}_k, \hat{\bar{c}}_k)$ represent the output of NBA at time k and for each sensor $S_i, i \in \mathcal{I}_k$.

It is well known that the EKF is prone to diverge, mainly for bad initial estimates and high noises [6], but no testable convergence conditions are known to our knowledge.

It is worth to remind that, differently from the linear case, the trace of $P_{k|k}$ only is an approximation (because of the linearization) of the MS estimation error $E\{\|x_k - \hat{x}_{k|k}\|^2\}$.

4. THE SENSORS SWITCHING POLICY

Choosing at any time instant to activate q out of p sensors ($q \ll p$) is an old problem; its original motivation, more than two decades ago, was the limited capacity of the transmission medium and the low computational power of the processors, which imposed to select at any instant only a few sensors.

Nowadays the above difficulties are vanished, thanks to the huge developments in the transmission and computational devices; however, a new and important problem is the need to save the lifetime of the batteries that power the sensors.

In such a contest it is very meaningful to device a policy that uses a small fraction of the available sensors to limit power consumption. It is evident that such a policy has an impact on the quality of the estimate, and that the best policy is the one that, for a fixed number $q \ll p$ of sensors, gives the best, in some statistical sense, estimate.

To formalize the problem, suppose that at each instant we activate q out of p sensors, to which a q-valued output equation $h(\cdot)$ and a $q \times n$ output matrix C_k correspond by Eq. (6) and (8). Different choices of the activation sequence clearly return different estimates of x_k for the EKF and we are interested in finding the best, in some statistical sense, estimate.

The switching policy we propose is the following:

> *Among the p sensors, choose q of them in such a way that the effect of the current observation on the reliability of the estimate is maximized.*

In principle q could depend on k, but here we will assume it constant for simplicity.

In a linear estimation problem a meaningful measure of the estimate reliability is the trace of the estimation error covariance matrix $P_{k|k}$, which corresponds to the MS estimation error, and a simple way to evaluate the effect of the current observation y_k on the a-posteriori covariance is to consider the trace of the difference between $P_{k|k-1}$ and $P_{k|k}$, i.e., the a-priori and a-posteriori estimation error.

In a nonlinear filtering problem the matrix $P_{k|k}$ is just an approximation of that covariance; nevertheless its trace has been chosen as our criterion by analogy with the linear case, and because it is simple to compute, as we will show.

Although there is no guarantee that this choice may take to the optimal switching sequence (finding it would be a problem of combinatorial complexity), we look at this criterion as a heuristic method to improve the convergence properties of the EKF.

Computing the trace criterion for the EKF

The difference $J_k = P_{k|k-1} - P_{k|k}$, whose trace has to be maximized by the choice of C_k, is simply

$$J_k = P_{k|k-1} C_k' (C_k P_{k|k-1} C_k' + V)^{-1} C_k P_{k|k-1};$$

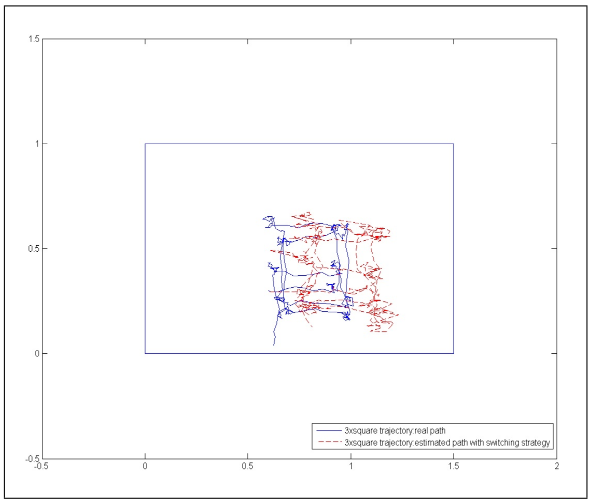

Fig. 3. "simple" rectangular world (*worldA*); 3 × square trajectory: real and estimated with switching strategy

when a small fraction of sensors is used, the computational cost of this task is not high, because of the many common terms in the formula, and it is suitable to be done online.

The approach of minimizing at each step the trace of J_k has been introduced first in [8] for the state estimation of linear systems, and then extended to the EKF [9] for the estimation of the unknown parameter in a Distributed Parameter System.

By minimizing J_k index at each time step k, we obtain the sensors index set $\mathcal{I}_k$ to be used in the NEKF algorithm.

5. NUMERICAL SIMULATIONS

To evaluate the performances of the proposed filter and of the sensor switching rule we have performed three series of experiments, each refereeing to a different trajectory followed by the robot. We tested the NEKF on the worlds illustrated in Figure 3 and 4. The first one, denoted by *worldA*, is a "simple" rectangular world of $1.5 \times 1.0m$. The second one, denoted by *worldB*, is a more "complex" world not comparable to any plane geometric Figure. Each trajectory is internal to these worlds.

The first trajectory is a square of 0.40 m per side, starting from $(0.60, 0.30)$. The second one is a square of 0.40 m per side, starting from $(0.60, 0.30)$, repeated three times. The third one is an hourglass trajectory starting from $(0.60, 0.30)$ and passing by $(1.1, 0.30)$, $(0.60, 0.60)$, $(1.1, 0.60)$, $(0.60, 0.30)$. The robot runs the three closed trajectories moving in counterclockwise sense. For each of them, we have performed 20 simulations, with different state and output noises realizations.

A sample time $T = 1$ second has been taken. The first trajectory is completed in $k_f = 120$ seconds, the second one in $k_f = 380$ seconds and the third one in $k_f = 180$ seconds. For each trajectory we impose a trapezoidal profile to the angular velocities of the wheels.

In each simulation, we have tested NEKF in three different situations. The first one uses a single sensor s_j, keeping it fixed along the path ($I_k = \{j\}, k \in \{0, \ldots, k_f\}$); this has been repeated for all sensors. The second uses one sensor ($q = 1$) out of the five available, switching between sensors with the switching policy discussed in the previous section (I_k is chosen at each step to minimize the index J_k

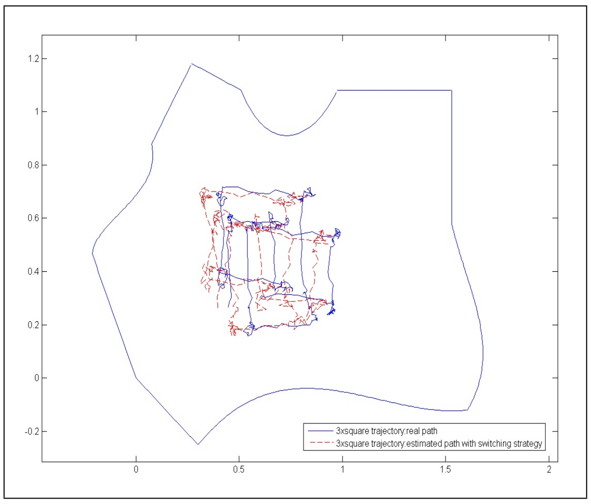

Fig. 4. "complex" world (*worldB*); 3 × square trajectory: real and estimated with switching strategy

presented in section 4). The third one uses all five sensors together ($I_k = \{1, \ldots, 5\}, k \in \{0, \ldots, k_f\}$), and it is used for reference.

The following values for the covariance parameters and the initial conditions were common to all simulations:

- $W = 10^{-2}\text{diag}\{15, 15, 0.004569\}$, which corresponds to a standard deviation of 0.003 m on x^1 and x^2 and of 0.38° on θ.
- $V = 4 \times 10^{-4} I$; this means a standard deviation of 0.02 m for each sensor, supposed having the same statistical properties.
- $\hat{x}_{0|0} = [0.65, 0.35, 0.01]'$; the true initial state of the robot is $x_0 = [0.6, 0.3, 0]'$ for each trajectory.
- $P_{0|0} = \text{diag}\{0.0025, 0.0025, 0.0027\}$, which corresponds to a standard deviation of 0.05 m on the robot position and 3° on the orientation.

The NBA requires as a parameter the radius r illustrated in previous sections. We adopted $r = 0.1$ m in all simulations done.

To evaluate the performance of the filter we introduce:

$$\varepsilon = \frac{1}{k_f + 1} \sum_{k=0}^{k_f} \sqrt{(x_k^1 - \hat{x}_{k|k}^1)^2 + (x_k^2 - \hat{x}_{k|k}^2)^2},$$

$$\mu = \frac{1}{k_f + 1} \sum_{k=0}^{k_f} \text{tr}\{P_{k|k}\},$$

where the first index is a measure of the localization error, and the second a measure of the estimate reliability.

We report in Tables 1 to 2 the values of the indexes for all trajectories, averaged over the 20 experiments, tested on *worldA* and in Tables 3 to 4 the values of the indexes for all trajectories tested on *worldB*.

In all the tables:

- s_j refers to the case where the j-th sensor is kept fixed along all the path
- *Switching* refers to the proposed switching policy
- *All* refers to the case where all sensors are used together. This is reported for the sake of comparison.
- *Square* refers to the first trajectory.
- 3 × *square* refers to the second trajectory.
- *Hourglass* refers to the third trajectory.

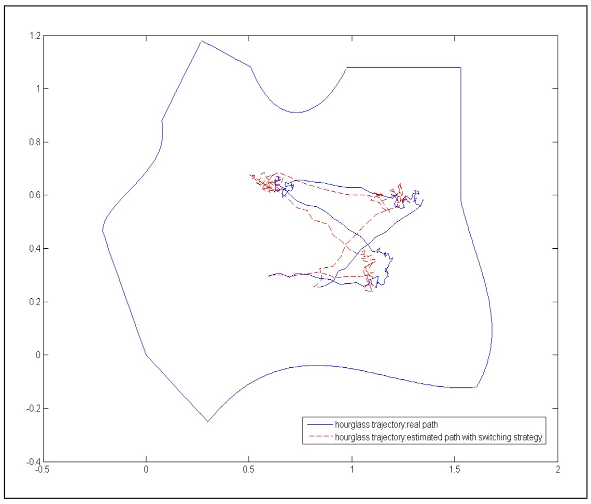

Fig. 5. Hourglass trajectory: real and estimated with switching strategy in "complex" world

6. CONCLUSIONS AND FUTURE WORKS

The experiments have shown that the proposed switching rule is effective in giving a more reliable estimate of the robot pose. Specifically, Tables 1 and 3 show that relying on one switching sensor is more effective than rely on one fixed sensor, whichever it is. Tables 2 and 4 show that the μ index is minimized more in the switching sensor policy than in the fixed sensor policy. All results obtained show that the NEKF improve the outcome independently from the environment configuration.

Currently the algorithm does not provide an environment mapping however all the information needed to build such a map is already available; we are developing solutions to provide a reliable mapping given such information.

Results shown in Tables 1 to 4 were obtained by simulation. We are going to validate the filter experimentally. Preliminary results are very encouraging and show that the filter works well regardless of the path traveled by the robot and the environment in which it moves.

A further aspect that we want to consider is loss of information on the data transmission channel. We will consider the NEKF outcome also in this situation.

REFERENCES

1. V. Gupta, T. Chung, B. Hassibi, R. M. Murray, On a stochastic sensor selection algorithm with applications in sensor scheduling and dynamic sensor coverage, *Automatica*, 42(2):251–260, 2006
2. V. Crugliano, L. D'Alfonso, P. Muraca, P. Pugliese, Experimental results of a sensor switching policy for mobile robots, *Proc. of the 18th Mediterranean Conf. on Control & Automation*, Marrakech, Morocco, June 2010, pp. 581–585
3. E. Ivanjko, I. Petrović, Extended Kalman Filter based mobile robot pose tracking using occupancy grid maps, *Proc. of the IEEE MELECON*, Dubrovnik, Croatia, May 2004, pp. 311–314
4. B. D. O. Anderson, J. B. Moore, *Optimal Filtering*, Prentice-Hall, 1979
5. M. S. Grewal, A. P. Andrews, *Kalman Filtering: theory and practice using MATLAB*, 3rd ed., Wiley, 2008

Table 1. NEKF - Index ε in meters - *worldA*

Trajectory	s_1	s_2	s_3	s_4	s_5	*Switching*	*All*
Square	0.117	0.138	0.137	0.171	0.178	0.120	0.115
$3 \times$ *square*	0.121	0.128	0.186	0.205	0.177	0.125	0.113
Hourglass	0.115	0.153	0.150	0.160	0.173	0.123	0.099

Table 2. NEKF - Index μ*100 - *worldA*

Trajectory	s_1	s_2	s_3	s_4	s_5	*Switching*	*All*
Square	0.107	0.104	0.107	0.099	0.114	0.051	0.019
$3 \times$ *square*	0.113	0.111	0.117	0.114	0.132	0.051	0.020
Hourglass	0.115	0.098	0.101	0.101	0.108	0.051	0.020

Table 3. NEKF - Index ε in meters - *worldB*

Trajectory	s_1	s_2	s_3	s_4	s_5	*Switching*	*All*
Square	0.107	0.125	0.097	0.159	0.124	0.090	0.074
$3 \times$ *square*	0.133	0.157	0.159	0.196	0.143	0.105	0.082
Hourglass	0.109	0.137	0.118	0.118	0.124	0.092	0.077

Table 4. NEKF - Index μ*100 - *worldB*

Trajectory	s_1	s_2	s_3	s_4	s_5	*Switching*	*All*
Square	0.101	0.102	0.103	0.088	0.107	0.050	0.018
$3 \times$ *square*	0.103	0.103	0.105	0.104	0.119	0.050	0.019
Hourglass	0.105	0.097	0.094	0.092	0.102	0.052	0.019

6. K. Reif, S. Günther, E. Yaz, R. Unbehauen, Stochastic stability of the discrete-time extended Kalman filter, *IEEE Trans. Automatic Contr.*, 44(4):714–729, 1999
7. The Mathworks, Inc., *MATLAB User's Guide*, Natick, MA, 1996
8. L. Carotenuto, P. Muraca, G. Raiconi, On the optimal design of the output transformation for discrete-time linear systems, *J. Optim. Theory Appl.*, 68(1):1–18, 1991
9. P. Muraca, P. Pugliese, An adaptive filter using scanning observations with applications to a DPS, *Int. J. Adaptive Contr. Signal Proc.*, 8(6):543–551, 1994
10. P. Muraca, P. Pugliese, G. Rocca, Convergence analysis of an extended Kalman filter using sensor querying and intermittent observations, *Proc. of the European Control Conf.*, Budapest, Hungary, August 2009, pp. 1414–1419
11. S. Thrun, W. Burgard, D. Fox, *Probabilistic Robotics*, MIT Press, 2005
12. J. Leonard, H. Durrant-Whyte, Mobile Robot Localization by tracking geometric beacons, *IEEE Trans. Robotics Autom.*, 7(3):376–382, 1991
13. H. Durrant-Whyte, T. Bailey, Simultaneous localisation and mapping (SLAM): part I - The essential algorithms, *IEEE Robotics and Automation Magazine*, vol. 2, 2006
14. B. Hiebert-Treuer, An introduction to robot SLAM (Simultaneous Localization And Mapping), (Available on `http://scholar.google.com`)
15. S. Riisgaard, M.R. Blas, SLAM for dummies, a tutorial approach to simultaneous localization and mapping, (Available on `http://ocw.mit.edu/`)