

Interlaced Extended Kalman Filter for Real Time Navigation

Stefano Panzieri and Federica Pascucci
Dip. di Informatica e Automazione
Università Roma Tre
Via della Vasca Navale 79, 00146, Roma, Italy
{panzieri,pascucci}@dia.uniroma3.it

Roberto Setola
Lab. Sistemi Complessi & Sicurezza
Università CAMPUS Bio-Medico
Via E. Longoni 86, 00153, Roma, Italy
r.setola@unicampus.it

Abstract—Real-time applications ask for reduced computational cost algorithms. In robotic exploration of unstructured environments the problem is more challenging: several tasks, at the same time, must be carried on ranging from reactive behaviours to the building of a structured representation of the environment itself. Many sensor signals have to be processed at each step to estimate both landmarks and robot positions. This mapping aptitude can be implemented through an Extended Kalman Filter recently proposed in a previous paper. Due to the large number of estimated variables, and real-time constraints, the filter is better implemented in its interlaced version. The novelty of this paper consists in extending the IEKF filter, removing some hypothesis on the linearity of both state transition and observation mapping, in order to further reduce computational burden and then achieve a better trade-off among computational load and accuracy.

Index Terms—Mobile robotics, SLAM, Sensor fusion

I. INTRODUCTION

Mobile robot operating in real world environments run several software modules. Among the others, we find the manager of the map of the area, and a localisation subsystem. The implemented control architecture includes, very often, an *a priori* knowledge of this map, as accurate as possible, in view of the accomplishment of the requested tasks. In this framework, the localisation subsystem must have the profitable skill of providing a reliable estimation of vehicle position.

The situation is slightly different when both the map and the robot location are not available from the beginning. In this case, starting from an unknown position, the robot will try to explore the environment using its sensors, to incrementally build an internal representation of the world containing features of both the environment and the robots itself. The map, build by the robot, is used back to compute its pose in the environment [1]. This problem is referred in literature as simultaneous localisation and map building (SLAM), and several approaches have been investigated to solve it after the seminal paper of Smith and Cheeseman [8]. The solution of SLAM has been addressed with approaches based on Bayesian filtering [4], [1]. These techniques approximate the probability representation using samples of

probability density distributions [9]. Although they are still computationally expensive for real time applications, they present significant advantages in solving the data association problem.

The approach that is more promising, from the implementation point of view, is definitely the one that uses the well-known predictor-corrector structure of the Kalman Filter. Assuming Gaussian distributions [2] for the errors we have to build a state-space model including the robot pose and the landmark positions. Measures coming from the proprioceptive sensory system (e.g., encoders, gyro) are used to feed the prediction step and produce a raw world estimation, while, the exteroceptive sensors (e.g., range finders, vision system) are used to refine the estimation producing a new map and robot location.

Drawbacks of such approach are the memory requirements and computational loads that quadratically increase with the number of map objects (*beacons*). In densely populated environments, the number of beacons detected will make those needs to be beyond the power capabilities of computer resources.

In this paper, we propose a solution to SLAM based on the Interlaced version of the Extended Kalman Filter (IEKF) [3] that appears suitable for real-time implementations. Indeed, this algorithm has a memory occupancy and a computational load linearly proportional to the number of beacons detected by the sensory system. IEKF has been proposed by the authors for the SLAM problem in [6] where experimental data have shown the good trade-off between accuracy and computational load obtained via that filter. However, using a generalised version of IEKF an excellent performance will be achieved, obtaining, as will be shown, a better convergence due to the preservation of more coupling effects than version reported in [6] has.

The algorithm has been experimentally tested using a robotised wheelchair equipped with a vision system and using ceiling lamps as natural landmarks. A brief description of the image processing algorithm will be given in Section II. In Section III we review the interlaced Kalman filter introduced in [3], while in Section IV and V we describe respectively

its application to the SLAM case, and the associated computational costs. Experimental results shown in Section VI conclude the paper.

II. VISION SYSTEM

As stated above, the vision system integrated in the robotised wheelchair should be able to recognise artificial sources of light during the navigation in an office like environment.

A low cost camera (a web cam) is mounted focusing the ceiling such that the distance between the image plane and the landmarks along the focal axis is fixed and known. This reduce the overall complexity of features extraction, because also the size (i.e., the area) of the landmarks is fixed and known.

The image processing algorithm implemented follows the fundamental steps of image analysis: preprocessing, segmentation, feature extraction.

- Preprocessing

Assuming that the beacon have a frame representation with large luminance values, a threshold is applied to obtain a binary image. The key function of the threshold operation is to improve the image in way that increases chances for the success of other processes: indeed, in this step small reflection are partially removed cause their areas normally decrease.

- Segmentation

After thresholding, the image is partitioned into its constituents objects, performing a graph search of all connected components (*blobs*). The segmentation step usually retrieves more than one blob, and, sometimes reflections can produce small connected components that can be discriminated evaluating their area.

- Feature extraction

After segmentation the position of the lamps in the image plane is obtained applying the well known formulas for calculating the centre of mass. In order to improve the quality of this shape, a binary morphological operator is applied on the connected components. An opening operation with a circular structuring element is performed to smooth the contour and eliminate protrusions.

III. IEKF FILTER

The Interlaced Extended Kalman Filter (IEKF) has been proposed in [3] to reduce computational load of the estimation process for a class of nonlinear system. The fundamental idea of the IEKF 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. In the context of estimation,

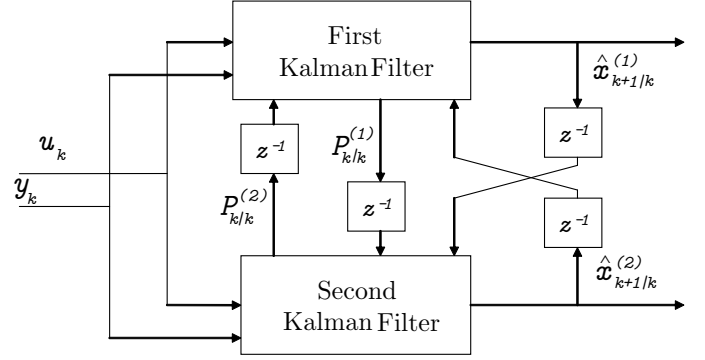


Fig. 1. Interlaced Extended Kalman Filter

the players are the estimation algorithm, the strategy is the estimate, the object function is a measure of the covariance estimation error. In particular IEKF consists of m parallel implementations of Kalman Filters (KF). Each KF works independently by the others and is designed to estimate a subset of the state variables, considering the remaining parts as deterministic time varying parameters. The error introduced is partially alleviated increasing the noise covariance matrices, as explained later and in [3].

For sake of simplicity, let us consider a system whose state vector can be partitioned into just two subsets, and can be put in the form

$$\begin{bmatrix} x_{k+1}^{(1)} \\ x_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} A^{(1)} & \mathbf{0} \\ \mathbf{0} & A^{(2)} \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} + \begin{bmatrix} f^{(1)}(x_k^{(2)}, u_k) \\ f^{(2)}(x_k^{(1)}, u_k) \end{bmatrix} + \begin{bmatrix} \xi_k^{(1)} \\ \xi_k^{(2)} \end{bmatrix} \quad (1)$$

where the state vector $x \in \mathcal{R}^n$ has been partitioned into $x^{(1)} \in \mathcal{R}^{n_1}$ and $x^{(2)} \in \mathcal{R}^{n_2}$ (with $n = n_1 + n_2$), $f^{(i)}$ are differentiable functions, and $\xi_k^{(i)} \in \mathcal{R}^{n_i}$, $i = 1, 2$ are zero-mean uncorrelated white process noise vectors characterized by the covariance matrices $Q_k^{(i)}$, $i = 1, 2$ and u_k is the input vector.

Further, the system is assumed to have an output equation that can be put into the following equivalent forms [3].

The output equation of the system

$$y_k = C^{(1)}(x_k^{(2)})x_k^{(1)} + D^{(1)}(x_k^{(2)}) + \psi_k \quad (2a)$$

$$y_k = C^{(2)}(x_k^{(1)})x_k^{(2)} + D^{(2)}(x_k^{(1)}) + \psi_k \quad (2b)$$

where $\psi_k \in \mathcal{R}^m$ is a zero-mean uncorrelated white measurement noise vector characterized by the covariance matrices R_k .

Under these hypothesis, the $x^{(1)}$ dynamic can be considered as a linear system depending on the time-varying parameter $x^{(2)}$, i.e., the estimate $\hat{x}_{k|k}^{(1)}$ is computed using the predictive estimate $\hat{x}_{k|k-1}^{(2)}$ obtained, at the previous step, by the other filter. Indeed, after replacing $x_k^{(2)}$ with $\hat{x}_{k|k-1}^{(2)}$ in (1) the first subsystem can be considered as a linear time varying system dependent on the known input $f^{(1)}(\hat{x}_{k|k-1}^{(2)}, u_k)$. At the same time, the output equation 2a turns out to be a linear equation whose matrices $C^{(1)}$ and $D^{(1)}$ depends on time-varying parameters. Similar considerations holds for the second subsystem.

Each KF is characterised by the following equations (for the first filter $i = 1$ and $j = 2$, while for the second $i = 2$ and $j = 1$):

$$\tilde{Q}_k^{(i)} = Q_k^{(i)} + J_{x,j}^{f,i} P_{k-1|k-1}^{(j)} J_{x,j}^{f,i T} + J_{u,j}^{f,i} Q_u^{(j)} J_{u,j}^{f,i T} \quad (3a)$$

$$P_{k|k-1}^{(i)} = A^{(i)} P_{k-1|k-1}^{(i)} A^{(i) T} + \tilde{Q}_k^{(i)} \quad (3b)$$

$$\tilde{R}_k^{(i)} = R_k + C^{(j)}(\hat{x}_{k|k-1}^{(i)}) P_{k|k-1}^{(j)} C^{(j)}(\hat{x}_{k|k-1}^{(i)})^T \quad (3c)$$

$$K_k^{(i)} = P_{k|k-1}^{(i)} C^{(i)}(\hat{x}_{k|k-1}^{(j)})^T [C^{(i)}(\hat{x}_{k|k-1}^{(j)}) P_{k|k-1}^{(i)} \times C^{(i)}(\hat{x}_{k|k-1}^{(j)})^T + \tilde{R}_k^{(i)}]^{-1} \quad (3d)$$

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_k^{(i)} [y_k - C^{(i)}(\hat{x}_{k|k-1}^{(j)}) \hat{x}_{k|k-1}^{(i)} + D^{(i)}(\hat{x}_{k|k-1}^{(j)})] \quad (3e)$$

$$\hat{x}_{k+1|k}^{(i)} = A^{(i)} \hat{x}_{k|k}^{(i)} + f^{(i)}(\hat{x}_{k|k-1}^{(j)}, u_k) \quad (3f)$$

$$P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} C^{(i)}(\hat{x}_{k|k-1}^{(j)}) P_{k|k-1}^{(i)} \quad (3g)$$

where $J_{x,j}^{f,i}$ is the Jacobian of $f^{(i)}$ with respect to $x^{(j)}$, $P_{k|k-1}^{(i)}$ is the covariance matrix of the estimation error variable $e_{k|k-1}^{(i)} := x_k^{(i)} - \hat{x}_{k|k-1}^{(i)}$ for $i = 1, 2$.

From (3a) and (3c) one can notice that the process and measurement noise covariance matrices $Q_k^{(q)}$ and R_k are suitable increased by addition of positive semi definite quantities that take into account the error introduced by the decoupling operation. Indeed, $\tilde{Q}_k^{(i)}$ and $\tilde{R}_k^{(i)}$ represent, respectively, the process and measurement covariance of interlaced errors, and their components are updated as

$$\tilde{\xi}_k^{(i)} := \xi_k^{(i)} + J_{x,j}^{f,i} e_{k|k-1}^{(i)} + J_{x,j}^{f,i} u_k \quad (4)$$

$$\tilde{\psi}_k^{(i)} := \psi_k^{(i)} + C^{(j)}(\hat{x}_{k|k-1}^{(i)}) e_{k|k-1}^{(i)}. \quad (5)$$

This formulation of IEKF assumes that state transition mapping and the observation mapping depend both linearly and affinely on their arguments. If one removes these assumptions, the algorithm can be still applied by linearising, at each step, each part of the system in the neighbourhood of the estimations obtained by the other filters at the previous step.

IV. IEKF SLAM FILTER

In this paper we use IEKF filter to solve the SLAM problem. In this framework we assume that all uncertainty sources have unimodal Gaussian distribution and provide a model for the robot, the beacon positions, and the sensors. The mobile platform considered is a robot with the kinematics of an unicycle. The robot is equipped with encoders and gyro, as proprioceptive sensors, while uses the vision system presented above as exteroceptive sensor. The measurement provided by the exteroceptive sensory system are expressed in the robot coordinates and represent the position of the beacons in the viewing windows of the web cam. Under the SLAM framework, the filter is able to localise the robot and concurrently build a simple geometric map (a list of beacon positions). During the navigation task, the system detects new features when exploring new areas. Once those features become reliable, they are included into the map.

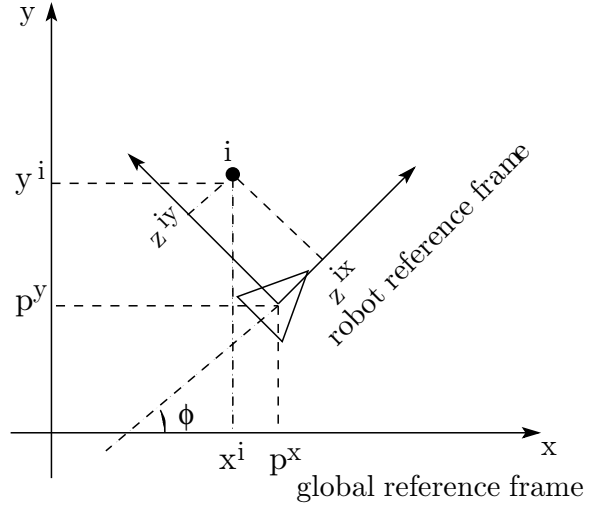


Fig. 2. Reference frames used in SLAM filter

A. Complete system model

The state of the whole system at the k -th sampling interval is composed by the configuration of the robot together with the positions of all the discovered beacons with respect to a global reference frame (see Fig. 2):

$$x_k = (x_k^r T, x_k^b T)^T. \quad (6)$$

Define the robot state vector as

$$x_k^r = (p_k^x, p_k^y, \phi_k, b_k)^T \quad (7)$$

where b is the gyro bias, and define the inputs for the robot model as

$$u_k = (\delta s_k, \omega_k)^T \quad (8)$$

where δs_k is the vehicle displacement and ω_k its angular velocity during the k -th sampling interval.

The robot dynamic is modelled using the equation of the unicycle model:

$$\begin{aligned} x_k^r &= f(x_{k-1}^r, u_k) = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\delta t_k \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1}^r + \begin{bmatrix} \cos \tilde{\phi}_k & 0 \\ \sin \tilde{\phi}_k & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} u_k \end{aligned} \quad (9)$$

where $\tilde{\phi}_k = \phi_{k-1} + (\omega_k - b_{k-1}\delta t_k)/2$ is the average robot orientation during the sampling time interval δt_k .

The beacon state vector is defined as

$$x_k^b = [p_k^1, \dots, p_k^N]^T \quad (10)$$

where $p_k^i = (x_k^i, y_k^i)$ is the location of the i -th beacon in the global reference frame. The state transition equation for the beacons can be written as

$$x_k^b = x_{k-1}^b \quad (11)$$

since beacons are assumed to be static. Notice that the size of x_k^b is dynamically increased any time a new beacon appears in the camera image.

The observation equation describes the relation between robot configuration and position of beacons in the viewing windows of the web cam (referred as *active beacon* in from now on). The observation vector

$$z_k = h(x_k) \quad (12)$$

consists of sub vectors z_k^i , $i = 1, \dots, M$, where M is the number of active beacons and

$$z_k^i = h^i(x_k) = \begin{bmatrix} z_k^{i,x} \\ z_k^{i,y} \\ z_k^i \end{bmatrix} = \mathbf{R}_k^\phi \begin{bmatrix} x_k^i \\ y_k^i \end{bmatrix} - \mathbf{R}_k^\phi \begin{bmatrix} p_k^x \\ p_k^y \end{bmatrix} \quad (13)$$

being \mathbf{R}_k^ϕ the rotation matrix between the robot reference frame and the global frame reference (see Fig. 2)

$$\mathbf{R}_k^\phi = \begin{bmatrix} \cos \phi_k & \sin \phi_k \\ -\sin \phi_k & \cos \phi_k \end{bmatrix} \quad (14)$$

This system is such that, as shown in [6], can be partitioned into $2 + M$ subsystem:

- *robot position subsystem* related to (p_k^x, p_k^y)
- *robot orientation subsystem* related to (ϕ_k, b_k)
- M *beacon position subsystems* each one related to a beacon position $p_k^i = (x_k^i, y_k^i)$

It is easy to recognise that partitioning in this way the state, each subsystem has dynamic and observation equations depending affinely with respect to its part of the state vector (in Section VI, to experimentally compare performance, we referred to this formulation as IEKF-1).

However, intensive numerical simulations have shown that better performance are achieved considering a different partitioning for the state vector. Specifically, considering a subsystem devoted to estimation robot state x^r , and M subsystems each one devoted to estimate the position of any single beacon (this algorithm will be referred as IEKF-2, in order to distinguish it from that proposed in [6]).

B. Robot subsystem

As mentioned before, the robot dynamics is described by

$$x_k^r = f(x_{k-1}^r, u_k) + \xi_{k-1}^r \quad (15)$$

where f is the unicycle equation (9) and $\xi_k^r \in \mathcal{R}^4$ is a zero-mean white noise vector with covariance matrix Q_k^r .

Because f is a nonlinear mapping, in order to apply IEKF we need to linearise it, obtaining

$$J_{x,r}^{f,r} = \begin{bmatrix} 1 & 0 & -\delta s_k \sin(\tilde{\phi}_k) & \frac{1}{2} \delta s_k \sin(\tilde{\phi}_k) \delta t_k \\ 0 & 1 & \delta s_k \cos(\tilde{\phi}_k) & -\frac{1}{2} \delta s_k \cos(\tilde{\phi}_k) \delta t_k \\ 0 & 0 & 1 & -\delta t_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

where $\tilde{\phi}_k = \phi_{k/k-1} + (\omega_k - \hat{b}_{k/k-1}\delta t_k)/2$. This quantity will be used, as in a classical EKF, in the estimation of the covariance matrix during innovation process, i.e., equation (3b), instead of $A^{(i)}$.

Moreover, we need to compute also the Jacobian of $f(\cdot)$ with respect to u

$$J_u^{f,r} = \begin{bmatrix} \cos(\tilde{\phi}_k) & -\frac{1}{2} \delta s_k \sin(\tilde{\phi}_k) \delta t_k \\ \sin(\tilde{\phi}_k) & \frac{1}{2} \delta s_k \cos(\tilde{\phi}_k) \delta t_k \\ 0 & \delta t_k \\ 0 & 0 \end{bmatrix} \quad (17)$$

The observation vector is composed by M sub vectors

$$z_k^i = h^i(x_k) = \mathbf{R}_k^\phi \begin{bmatrix} x_k^i \\ y_k^i \end{bmatrix} - \mathbf{R}_k^\phi \begin{bmatrix} p_k^x \\ p_k^y \end{bmatrix} \quad (18)$$

Due to the nonlinearity of the mapping, we have to evaluate the linearised version. However, for sake of brevity, we do not report the results. These linearised quantities are used, instead of matrix $C^{(i)}$, in equations (3c), (3d), and (3g). Obviously, in the state innovation equation (3f), and in the observation equation (3e), are implemented the associated nonlinear expressions.

C. Beacon position subsystem

The state transition model of each beacon is

$$p_k^{(i)} = p_{k-1}^{(i)} + \xi_{k-1}^{(i)}; \quad i = 1, \dots, M \quad (19)$$

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

The associated output equations result in

$$z_k^{(i)} = C^{(i)}(\phi_k)p_k^{(i)} + D^{(i)}(x_k^r) + \psi_k^{(i)} \quad (20)$$

where

$$C^{(i)}(\phi_k) = \mathbf{R}_k^\phi \quad (21)$$

$$D^{(i)}(x_k^r) = -\mathbf{R}_k^\phi p_k^{(i)}. \quad (22)$$

To simplify IEKF calculations, we pack, at each sampling time, the M beacon filters associated with active beacons into a single system. Specifically, at k -th sampling time, the active beacons are retrieved and each sub vectors is used to dynamically compose the state vector of such filter. For example, if the beacons i and j are in the viewing windows of the web cam, the active beacon subsystem become:

$$\begin{bmatrix} p_k^{(i)} \\ p_k^{(j)} \\ p_k \end{bmatrix} = \begin{bmatrix} p_{k-1}^{(i)} \\ p_{k-1}^{(j)} \\ p_{k-1} \end{bmatrix} + \begin{bmatrix} \xi_{k-1}^{(i)} \\ \xi_{k-1}^{(j)} \\ \xi_{k-1} \end{bmatrix} \quad (23)$$

$$\begin{bmatrix} z_k^{(i)} \\ z_k^{(j)} \\ z_k \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k^\phi & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k^\phi \\ \mathbf{0} & \mathbf{R}_k^\phi \end{bmatrix} \begin{bmatrix} p_k^{(i)} \\ p_k^{(j)} \\ p_k \end{bmatrix} - \begin{bmatrix} \mathbf{R}_k^\phi & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k^\phi \end{bmatrix} \begin{bmatrix} p_k^r \\ p_k^r \end{bmatrix}. \quad (24)$$

V. COMPUTATIONAL COST

From a computational point of view, this formulation has the same asymptotic behaviour of [6]. Specifically, it has a memory occupancy $\sim O(N)$, where N is the number of states used to represent all the discovered beacons and the vehicle configuration. The computational load, on the other way, linearly depends on the number of beacons simultaneously displayed inside camera image (i.e., M).

Notice that a very interesting feature able to overcome drawbacks of classical EKF based SLAM formulation is represented by this computational reduction.

Indeed, it is well known that classical SLAM algorithms have computational cost and memory requirement proportional to $O(N^2)$, being N the number of the states used to represent all the beacons and the whole vehicle position.

Different solutions have been proposed in literature to reduce this load. In [4] authors propose CSLAM, an algorithm that segments the map into a set of fixed disjointed areas and solves the SLAM problem only with respect to the beacons included inside one area (i.e., where the vehicle is). Full update of the filter is performed only when the vehicle moves away from the area. Further, to reduce memory occupation, in [5] the same authors introduce a de-coupling procedure in CSLAM that nullifies the correlation terms related with beacons belonging to different constellations. In this way, memory and computational requirements of the algorithm become proportional to $O(N \cdot N_b)$, where N_b is the number of beacons inside each area.

However, because $M \ll N$ (but also $M < N_b$), the proposed formulation shows a reduced computation load.

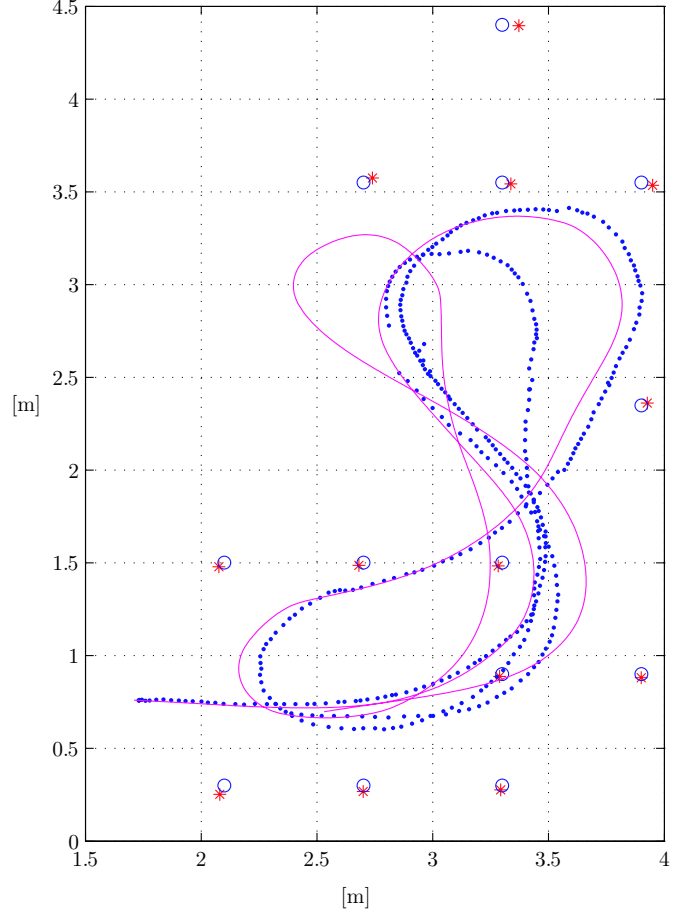


Fig. 3. Corrected path (dot), odometry (solid), and landmark estimated position (*) vs. real (o)

VI. EXPERIMENTAL RESULTS

Experimental trials have been carried out using a robotised wheelchair prototype built at the robotics lab of the University of “Roma Tre” and a Philips Vesta Pro Scan. The vehicle has two driving wheels equipped with low resolution incremental encoders (6.4 pulses/mm of the wheel movement).

The proprioceptive sensory system is completed by a piezoelectric gyro (MuRata), that measures rotation velocity. The gyro has a good accuracy (3%) but is affected by temperature depending bias.

The software implementation is based on two notebooks connected over an Ethernet link. The first laptop installed on the wheelchair runs the control software. A data acquisition card (DAQPad 1200 by National Instruments) interfaces the sensory and driving systems under a LabVIEW application that includes some C routines for the time critical tasks of

the filter implementation. The second laptop, is devoted to process images from the vision system including a web cam mounted on the robot and focusing the ceiling. The distance from the lights is about 2.50 m and each pixel is thereafter about 5mm large at the CIF resolution (352×288).

During the experiment the robot was driven in order to execute a double 8-path in an office-like environment. Using this set up the camera is able to view only few beacons (zero to three) for each frame.

The localisation results are shown in Fig. 3 where the odometric path, as estimated on the basis of encoders and gyro data, is compared with the filter output. The estimated beacons positions are shown in the same Figure as red stars and the distance from their exact position is reported in Fig. 4 (IEKF-2).

The data collected from the sensory system of the robot was processed off-line using three different algorithm: an approach based on classical Extended Kalman Filter (EKF), the IEKF proposed in [6] (IEKF-1) and the IEKF explained in Section IV (IEKF-2).

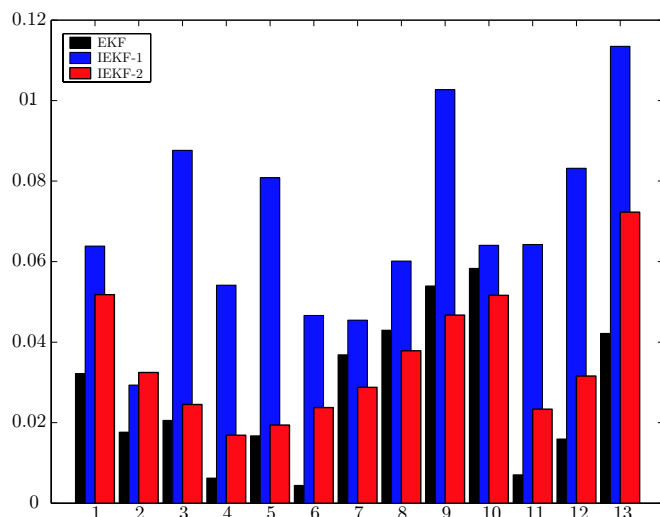


Fig. 4. Landmark estimation errors

In Table I we show some statistical values that characterize the quality of the three estimates.

TABLE I
MAP ERROR (CM)

	Mean value	Standard deviation	Min error	Max error	Execution Time
EKF	3.2	0.8	0.8	4.4	4.340
IEKF-1	5.5	1.14	3.8	7.2	2.604
IEKF-2	3.5	2.5	1.6	7.2	2.063

The proposed version IEKF-2 shows more accurate estima-

tion, very similar to that obtained via classical EKF but with a considerable reduction in computational load (compare the execution time of the different algorithms).

The best performance shown by the proposed formulation of IEKF with respect to IEKF-1 might be explained considering that those formulation “neglects” correlation among robot position and orientation (taking into account it only via the modified error covariance matrix \tilde{Q}). However, because these quantities are strongly related each other, neglecting their correlation represent a crude approximation. Best performances are obtained, obviously, using full EKF with a computational cost related to the update of the full state as already shown in [1].

Finally, we remark that, using the same experimental test bed, the execution time of the proposed filter is 60% less than the algorithm reported in [7].

VII. CONCLUSIONS

This paper describes an algorithm based on a modified version of EKF for SLAM problem. Specifically, the peculiar structure of the problem at hand allows the use of the interlaced version (IEKF) of EKF.

This algorithm represents a good trade-off between accuracy and computational load. In particular, it has been shown that computational load and memory capacity request by IEKF increase at least linearly with the number of beacons (while classical approaches have a quadratic dependency) . Further work are devoted to test IEKF in other configurations (outdoor), with different testbeds and in the presence of more mobile robots.

REFERENCES

- [1] Clark S., Dissanayake G., Newman P., and Durrant-whyte H.F. A solution to simultaneous localization and map building (slam) problem. *IEEE Trans. on Robotics and Automation*, vol. 17(no 3):229–241, 2001.
- [2] Gelb A.C. *Applied Optimal Estimation*. MIT Press, Cambridge,MA, 1994.
- [3] Glielmo L., Setola R., and Vasca F. An interlaced extended kalman filter. *IEEE Trans. on Automatic Control*, vol. 44(no 8):1546–1549, 1999.
- [4] Guivant J. and Nebot E. Optimization of simultaneous localization and map building algorithm for real time implementation. *IEEE Trans. on Robotics and Automation*, vol.17(no 3):242–256, 2001.
- [5] Guivant J. and Nebot E. Improving computational and memory requirements of simultaneous localization and map building algorithm. In *Proc. of 2002 IEEE Int. Conf. on Robotics and Automation*, Washington DC, 2002.
- [6] Panzieri S., Pascucci F., and Setola R. Simultaneous localization and map building algorithm for real-time applications. In *Proc. of 16th IFAC World Congress*, Praha, Czech Republic, 2005.
- [7] Panzieri S., Pascucci F., and Ulivi G. Vision based navigation using kalman approach for slam. In *Proc. of 11th Int. Conf. on Advanced Robotics*, Coimbra, Portugal, 2003.
- [8] Smith R.C. and Cheeseman P. On the representation and estimation of spatial uncertainty. *Int. J. of Robotics Research*, vol. 5(no 4):56–68, 1986.
- [9] Thrun S., Fox D., and Burgard W. A probabilistic approach to concurrent mapping and localization for mobile robots. *March learning Autonom Robots*, vol.31:29–53, 1998.