

## SIMULTANEOUS LOCALIZATION AND MAP BUILDING ALGORITHM FOR REAL-TIME APPLICATIONS

Stefano Panzieri \* Federica Pascucci \*  
Roberto Setola \*\*

\* Dip. Informatica e Automazione, Università “Roma Tre”, Via della Vasca Navale, 79, 00144, Roma, Italy

\*\* Università CAMPUS Biomedico - Facoltà di Ingegneria,  
Via E. Longoni, 86, 00153 Rome, Italy

**Abstract:** Mobile robot navigation in unstructured environment is a challenging task due to the uncertain nature of the real world. Navigating using visual landmarks could be a mandatory skill together with the ability of building a representation of the world around the robot. This mapping aptitude should be implemented as an efficient real-time task, even if a large number of elements have to be included in the map itself. To this aim, and to help in localising the robot, a promising technique is given by the Extended Kalman Filter in its interlaced version. The resulting SLAM algorithm, proposed in this paper, has a reduced computational cost preserving a good performance.

**Keywords:** Kalman filters, Mobile robots, Visual Motion, Real Time, Extended Kalman filters, SLAM

### 1. INTRODUCTION

When mobile robots operate in real world environments they require both reliable localisation systems, and a sufficiently precise map of the working area. Thus, most successful mobile robot systems include, up to date, a localisation module together with an *a priori* knowledge of the environment (i.e., a map) in their control architecture. With such a structure, and using the map, the localisation module is able to provide a reliable estimation of the position of the vehicle.

A more challenging problem arises when “both the map and the robot location are unknown. In this case, the robot starts in an unknown position, in an unknown environment, and tries to incrementally build a map while using the same map to compute its pose in the environment” (Clark S. *et al.*, 2001). 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 R.C. and Cheeseman P., 1986). The solution of SLAM has been addressed with approaches based on Bayesian filtering (Guivant J. and Nebot E., 2001; Clark S. *et al.*, 2001). These techniques approximate the probability representation using samples of probability density distributions (Thrun S. *et al.*, 1998). Although they are still computationally expensive for real time applications, they present significant advantages in solving the data association problem.

In this framework, one of the most appealing approaches to solving real time SLAM problems uses the well known predictor-corrector structure of the Kalman filter: a solution is given by modelling the environment and the sensors, and assuming that errors have a Gaussian distribution (Gelb

A.C., 1994). The measures of the proprioceptive sensory system of the vehicle (e.g., encoders, gyros) are used to compute a raw location estimation, while, the exteroceptive sensors refine this estimation and update the map.

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 Kalman based solution to SLAM, suitable for real-time implementations. Indeed, our algorithm has a memory occupancy and a computational load linearly proportional to the number of beacons detected by the sensory system. In particular, a vision system is interfaced with a mobile robot allowing the use of ceiling lamps as natural landmarks. A brief description of the image processing algorithm will be given in section 2. In section 3 we review the interlaced Kalman filter introduced in (Glielmo L. et al., 1999), while in section 4 we describe its application to the SLAM case. Its reduced computational cost is evaluated in section 5 and some experiments, whose results are shown in section 6, conclude the paper.

## 2. VISION SYSTEM

As stated above, the vision system presented in this work should be able to recognise artificial sources of light during the navigation of a mobile robot in an office like environment.

The camera 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 of the landmarks is fixed.

The image processing algorithm developed follows the fundamental steps of image analysis: pre-processing, 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.

## 3. IEKF FILTER

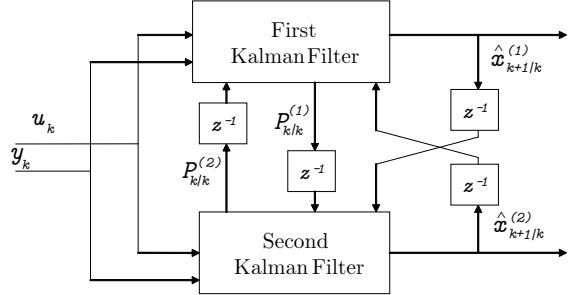


Fig. 1. Interlaced Extended Kalman Filter

The Interlaced Extended Kalman Filter (IEKF) has been proposed in (Glielmo L. et al., 1999) to reduce computational load of the estimation process for a class of nonlinear system. The fundamental idea of the IEKF 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 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 implementation 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.

For sake of simplicity, let us consider a system whose state vector can be partitioned into just two subsets, 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 form. 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  $\psi_k \in \mathcal{R}^m$ , are zero-mean uncorrelated white measurement noise vector characterized by the covariance matrices  $R_k$ .

Under this hypothesis, the first filter of the IEKF computes the estimate  $\hat{x}_{k|k}^{(1)}$  using the predictive estimate  $\hat{x}_{k|k-1}^{(2)}$  obtained, at the previous step, by the other filter. Notice that, after replacing  $x_k^{(2)}$  with  $\hat{x}_{k|k-1}^{(2)}$  in (1) and (2) the first subsystem can be considered as a linear time varying system dependent on the known input  $f^{(1)}(\hat{x}_{k|k-1}^{(2)}, u_k)$ . 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  $j = 1$ ):

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

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

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

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

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

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

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

where  $J_{f,j}^{x,i}$  is the Jacobian of  $f^{(i)}$  w.r.t.  $x^{(j)}$ ,  $P_{k|k-1}^{(i)}$  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 the “modified” error

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

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

IEKF assumes that state transition mapping and the observation mapping depend linearly and affinely on their argument. If one removes these assumptions, the presented 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.

#### 4. 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 feature become reliable, they are included into the map.

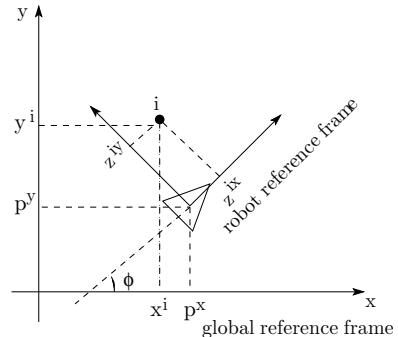


Fig. 2. Reference frames used in SLAM filter

##### 4.1 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 beacons w.r.t. a global reference frame (see Fig. 2):

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

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  $\delta s_k$  is the vehicle displacement and  $\omega_k$  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  $\delta t_k$ .

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

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

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

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

Due to the structure of the system model, IEKF can be adopted partitioning the system into  $2+M$  subsystem:

- *robot position subsystem* estimating  $p_k^r = (p_k^x, p_k^y)$
- *robot orientation subsystem* estimating  $\theta_k = (\phi_k, b_k)$
- $M$  *beacon position subsystems* each one estimating a beacon position  $p_k^i = (x_k^i, y_k^i)$

#### 4.2 Robot position subsystem

The robot dynamics can be written in the form

$$p_k^r = p_{k-1}^r + f^r(z_{k-1}, u_k, \delta t_k) + \xi_{k-1}^r \quad (15)$$

where  $\xi_{k-1}^r \in \mathcal{R}^2$  is a zero-mean white noise vector with covariance matrix  $Q_k^r$  and

$$f^r(\theta_{k-1}, u_k, \delta t_k) = \begin{bmatrix} \cos \tilde{\phi}_k & 0 \\ \sin \tilde{\phi}_k & 0 \end{bmatrix} u_k \quad (16)$$

The observation vector is composed by  $M$  sub vectors

$$z_k^{r,i} = C^r(\theta_k)p_k^r + D^r(p_k^{(i)}, \theta_k) + \psi_k^{(i)} \quad (17)$$

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

$$C^r(\theta_k) = -\mathbf{R}_k^\phi; \quad D^r(p_k^{(i)}, \theta_k) = \mathbf{R}_k^\phi p_k^{(i)}. \quad (18)$$

#### 4.3 Robot orientation subsystem

The orientation dynamic  $\phi$  and the gyro bias  $b$  may be formulated as

$$\theta_k = A^\theta \theta_{k-1} + f^\theta(u_k) + \xi_k^\theta \quad (19)$$

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

$$A^\theta = \begin{bmatrix} 1 & -\delta t_k \\ 0 & 1 \end{bmatrix}; \quad f^\theta = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} u_k \quad (20)$$

The output equation consists of  $M$  sub vector

$$z_k^{\theta,i} = h^\theta(\theta, p^r, p^{(i)}) + \psi_k^\theta \quad (21)$$

where  $h^\theta(\theta, p^r, p^{(i)})$  is a nonlinear mapping

$$h^\theta(\theta, p^r, p^{(i)}) = \mathbf{R}_k^\phi(p_k^{(i)} - p_k^r) \quad (22)$$

#### 4.4 Beacon position subsystem

The state transition model of each beacon is

$$p_k^{(i)} = p_{k-1}^{(i)} + \xi_{k-1}^{(i)}; \quad i = 1, \dots, M \quad (23)$$

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

$$z_k^{(i)} = C^{(i)}(\theta_k)p_k^{(i)} + D^{(i)}(p_k^r, \theta_k) + \psi_k^{(i)} \quad (24)$$

where

$$C^{(i)}(\theta_k) = \mathbf{R}_k^\phi; \quad D^{(i)}(p_k^r, \theta_k) = -\mathbf{R}_k^\phi p_k^r \quad (25)$$

In order to reduce the number of parallel subsystems, we decide to pack all the  $M$  beacon filters in the active beacon subsystem. At  $k$ -th sampling time the active beacon 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:

$$\begin{bmatrix} p_k^{(i)} \\ p_k^{(j)} \end{bmatrix} = \begin{bmatrix} p_{k-1}^{(i)} \\ p_{k-1}^{(j)} \end{bmatrix} + \begin{bmatrix} \xi_{k-1}^{(i)} \\ \xi_{k-1}^{(j)} \end{bmatrix} \quad (26)$$

$$\begin{bmatrix} z_k^{(i)} \\ z_k^{(j)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k^\phi & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k^\phi \end{bmatrix} \begin{bmatrix} p_k^{(i)} \\ p_k^{(j)} \end{bmatrix} - \begin{bmatrix} \mathbf{R}_k^\phi & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k^\phi \end{bmatrix} \begin{bmatrix} p_k^r \\ p_k^r \end{bmatrix} \quad (27)$$

#### 5. COMPUTATIONAL COST

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

In order to reduce the computational load in (Guivant J. and Nebot E., 2001) the authors propose CSLAM. This algorithm segments the map into a set of fixed disjointed areas and solves the SLAM problem only with respect to the beacons present inside one area (i.e. where there is the vehicle). Full update of the filter is performed only when the vehicle move away from the area. Further, to reduce memory occupation, in (Guivant J. and Nebot E., 2002), the authors introduce a de-coupling procedure in CSLAM nullifying the correlation terms related with beacons belonging to different constellations. In this way, memory and computational requirements of the algorithm are proportional to  $\sim O(N \cdot N_b)$ , where  $N_b$  is the number of beacons present inside to each area.

As noted before, our algorithm stores separately information related to each filter. Then it is im-

mediate to recognise that the memory occupancy is proportional to

$$M_R \sim 4 + 4 + 4 \cdot N$$

that can be approximated as  $M_R \sim O(N)$ .

On the other side, the computational load of the filters for position and orientation of the vehicle is proportional to  $\sim O(4 + 4 + M^2)$ , where  $M$  is the number of beacons present inside the camera image. However, notice that quite all the equations of the position filter, excepted that devoted to calculate the augmented process and measurement covariance matrices  $\tilde{R}_k^r$  and  $\tilde{Q}_k^r$ , have a cost proportional to a dimension four (the same also for the orientation filter neglecting the calculations requested to compute  $\tilde{R}_k^\theta$  and  $\tilde{Q}_k^\theta$ ).

During the estimation phase, the filters associated with the beacons present inside the camera image are assembled into a single object. The computational load of this filter is proportional to  $\sim O(4 + 4 + M^2)$ , but if we neglect the calculation requested to compute  $\tilde{R}_k^b$ , it is proportional to  $\sim O(M^2)$ .

In any case, because  $M \ll N$  (but also  $M < N_b$ ), we archive a considerable reduction of computation load.

#### 6. 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.

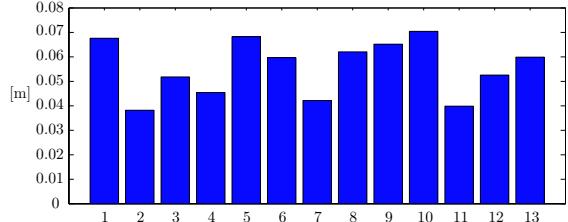


Fig. 3. Landmark estimation errors

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 (DAQ-Pad 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 ( $352 \times 288$ ).

The results our experiment are reported in Figs. 3 and 4. The robot describes a double 8-path in an office-like environment and the camera is able to view only few beacons (zero to three) for each frame. The estimated beacons positions are shown in Fig. 4 as red stars and the distance from their exact position is reported in Fig. 3. The statistical values reported in table 6 show that the quality of the estimate is almost the same for each beacon due to the decoupled nature of the filter.

The localisation results are shown in Fig. 4 where the odometric path, as estimated on the basis of encoders and gyro data, is compared with the filter output.

Table 1. Map error (in cm)

Mean value	Standard deviation	Min error	Max error
5.5	1.14	3.8	7

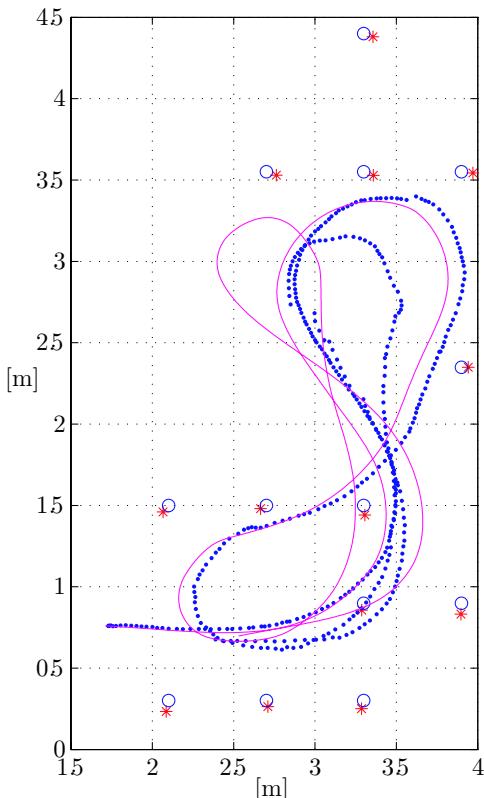


Fig. 4. Corrected path (blue), odometry (magenta), and landmark estimated position (\*) vs. real (o)

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 (Panzieri S. *et al.*, 2003).

## 7. CONCLUSIONS

The paper has demonstrated the feasibility of the IEKF approach to the SLAM problem. The particular formulation of the filter guarantees a reduced computational cost and lower memory requirement. Some experiments have shown the acceptable performance of the filter that turns out to be effective in office-like environments populated by a large number of beacons.

## REFERENCES

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