Robotica (2008) volume 26, pp. 817–830. © 2008 Cambridge University Press
doi:10.1017/S0263574708004414 Printed in the United Kingdom

A hybrid approach to fast and accurate localization for legged robots

Renato Samperio†*, Huosheng Hu†, Francisco Martín‡ and Vicente Matellán‡

†*Department of Computing & Electronic Systems, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, United Kingdom*
‡*Robotics Lab-GSyC, DITTE-ESCET-URJC, Universidad Rey Juan Carlos, Móstoles, Madrid, Spain*

(Received in Final Form: February 11, 2008. First published online: April 11, 2008)

SUMMARY
This paper describes a hybrid approach to a fast and accurate localization method for legged robots based on Fuzzy-Markov (FM) and Extended Kalman Filters (EKF). Both FM and EKF techniques have been used in robot localization and exhibit different characteristics in terms of processing time, convergence, and accuracy. We propose a Fuzzy-Markov–Kalman (FM–EKF) localization method as a combined solution for a poor predictable platform such as Sony Aibo walking robots. The experimental results show the performance of EKF, FM, and FM-EKF in a localization task with simple movements, combined behaviors, and kidnapped situations. An overhead tracking system was adopted to provide a ground truth to verify the performance of the proposed method.

KEYWORDS: Legged robots; robot localisation; Extended Kalman filters; Fuzzy-Markov methods.

1. Introduction

In general, mobile robot localization is about estimating a robot pose, i.e., position and orientation, relative to its environment. It is one of the fundamental problems in mobile robotics. Nowadays, there are many localization techniques that have been developed for mobile robots, including extended Kalman filtering, particle filtering, grid-based Markov localization algorithms, etc. However, no single localization method is perfect for diversified real-world applications. It is necessary to combine some of them in practical applications according to the different sensors being used.

Up to now, there has been a number of research works on hybrid methods which integrate probabilistic multi-hypothesis and grid-based maps with EKF-based techniques. Some methodologies as indicated in the literature[7,12] require an extensive computation but offer a reliable positioning system. By cooperating Markov into the localization process,[6] EKF positioning can converge faster with an inaccurate grid observation. In general, Markov-based techniques and grid-based maps[5] are classic approaches to robot localization and their computational cost is huge when the grid size in a map is small.[3,10] Therefore, the Monte Carlo (MCL) technique has partially solved this problem,[4,14,15] but still has difficulty handling environmental changes.[13] As a fast approach, EKF maintains a continuous estimation of robot position, but cannot manage multi-hypothesis estimations.[1]

It should be noticed that traditional localization techniques such as EKF perform position tracking, i.e., integrating the previous position of the robot with the robot motion prediction and new perception information. In fact, they are computationally efficient, but may fail when the estimated position is far from the true one. Therefore, it is difficult for them to be used for legged robots that have very poor odometry, such as leg slippage and loss of balance. As shown in ref. [8], this paper compares two already existing localization methods (EKF and FM) for Sony legged robots in a highly dynamic environment, i.e., the RoboCup soccer domain. Then, we propose a hybrid localization method that is a merge of both EKF and FM approaches. Such a merge results in reduced inconsistencies, fuses inaccurate odometry data with visual data as well as having less computational cost. The proposed FM–EKF localization algorithm implements a fuzzy cell to speed up convergence and maintains an efficient localization. Subsequently, the performance of the proposed method was tested in three experimental comparisons: (i) simple movements along the pitch, (ii) localising and playing combined behaviors, and (c) kidnapping the robot.

The rest of the paper is organised as follows. Section 2 presents an observer model for the Sony AIBO legged robot. The theoretical background of the proposed localization methods is outlined in Section 3. The system configuration is described in Section 4. Section 5 presents some experimental results to show the performance of three different localization methods. Finally, a brief conclusion and further research work are given in Section 6.

2. Observer Design

To begin with, robot motion is described with a state vector which contains three variables, i.e., 2D position (x, y) and

* Corresponding author. E-mail: renatosamperio@gmail.com

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

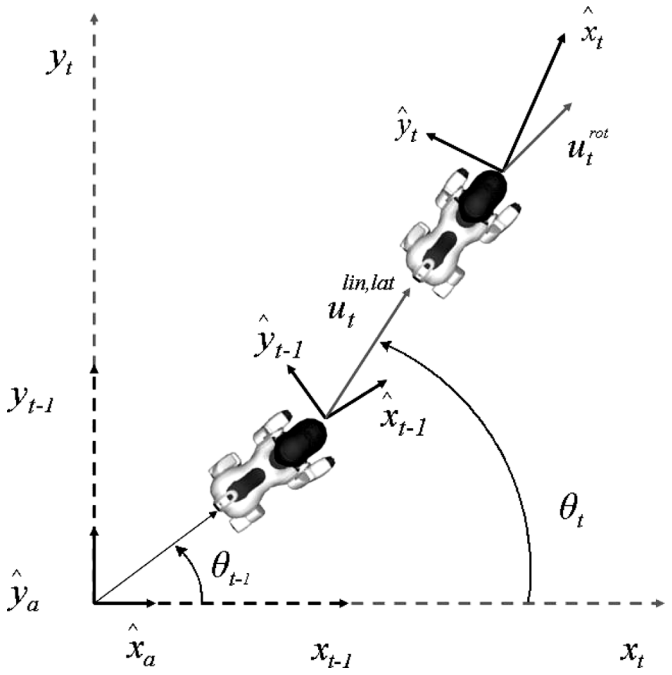


Fig. 1. The proposed motion model in this research.

orientation (θ).

$$\begin{pmatrix} x_t^- \\ y_t^- \\ \theta_t^- \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} \left(u_t^{lin} + w_t^{lin}\right)cos\theta_{t-1} - \left(u_t^{lat} + w_t^{lat}\right)sin\theta_{t-1} \\ \left(u_t^{lin} + w_t^{lin}\right)sin\theta_{t-1} + \left(u_t^{lat} + w_t^{lat}\right)cos\theta_{t-1} \\ u_t^{rot} + w_t^{rot} \end{pmatrix} \quad (1)$$

where u^{lat}, u^{lin}, and u^{rot} are the lateral, linear and rotational components of odometry, and w^{lat}, w^{lin}, and w^{rot} are the lateral, linear, and rotational components of odometry errors. $t-1$ refers to the previous step time and t to the current step.

In general, state estimation is a weighted combination between noisy states in both priori and posterior estimations. Assuming that v is the measurement noise and w is the process noise. We have the measurement noise covariance matrix R and the process noise covariance matrix Q as follows:

$$R = E[vv^t] \quad (2)$$

$$Q = E[ww^t] \quad (3)$$

Note that measurement noise in matrix R indicates sensor errors and Q matrix is a confidence indicator for the current prediction, where uncertainty for each state estimation can increase or decrease. We adopt an odometry motion model, u_{t-1} as f shown in Fig. 1. Table I describes all variables for three dimensional *(linear, lateral, and rotational)* odometry information.

According to the measured experimental data, we can see that odometry system has a deviation of 30% on average as shown in Eq. (4). Therefore, applying transformation matrix

Table I. Variable description for obtaining linear, lateral, and rotational odometry information.

Variable	Description
$\widehat{x}_a$	x axis of world coordinate system
$\widehat{y}_a$	y axis of world coordinate system
x_{t-1}	previous robot x position in world coordinate system
y_{t-1}	previous robot y position in world coordinate system
θ_{t-1}	previous robot heading in world coordinate system
$\widehat{x}_{t-1}$	previous state x axis in robot coordinate system
$\widehat{y}_{t-1}$	previous state y axis in robot coordinate system
$u_t^{lin,lat}$	lineal and lateral odometry displacement in robot coordinate system
u_t^{rot}	rotational odometry displacement in robot coordinate system
x_t	current robot x position in world coordinate system
y_t	current robot y position in world coordinate system
θ_t	current robot heading in world coordinate system
$\widehat{x}_t$	current state x axis in robot coordinate system
$\widehat{y}_t$	current state y axis in robot coordinate system

W_t from Eq. (5), noise can be declared as robot uncertainty where θ is the robot heading.

$$Q_t = \begin{pmatrix} \left(0.3u_t^{lin}\right)^2 & 0 & 0 \\ 0 & \left(0.3u_t^{lat}\right)^2 & 0 \\ 0 & 0 & \left(0.3u_t^{rot} + \frac{\sqrt{(u_t^{lin})^2 + (u_t^{lat})^2}}{500}\right)^2 \end{pmatrix} \quad (4)$$

$$W_t = \frac{\partial f}{\partial w} = \begin{pmatrix} cos\theta_{t-1} & -sen\theta_{t-1} & 0 \\ sen\theta_{t-1} & cos\theta_{t-1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5)$$

In addition, landmarks in the robot environment require notational representation for a feature vector f_t^i function in each i-th feature as in the following equation:

$$f(z_t) = \left\{f_t^1, f_t^2, \ldots\right\} = \left\{\begin{pmatrix} r_t^1 \\ b_t^1 \\ s_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ b_t^2 \\ s_t^2 \end{pmatrix}, \ldots\right\} \quad (6)$$

where landmarks are detected by an onboard active camera in terms of range r_t^i and bearing b_t^i; a signature s_t^i is added to represent each landmark. We use a landmark measurement model defined by a feature-based map m, which consists of a list of signatures and location coordinates as follows:

$$m = \{m_1, m_2, \ldots\} = \{(m_{1,x}, m_{1,y}), (m_{2,x}, m_{2,y}), \ldots\} \quad (7)$$

where the i-th feature at time t corresponds to the j-th landmark detected by a robot whose pose is $x_t = (x \;\; y \;\; \theta)^T$. we use the model:

$$\begin{pmatrix} r_t^i(x, y, \theta) \\ b_t^i(x, y, \theta) \\ s_t^i(x, y, \theta) \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ atan2(m_{j,y} - y, m_{j,x} - x)) - \theta \\ s_j \end{pmatrix} \quad (8)$$

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

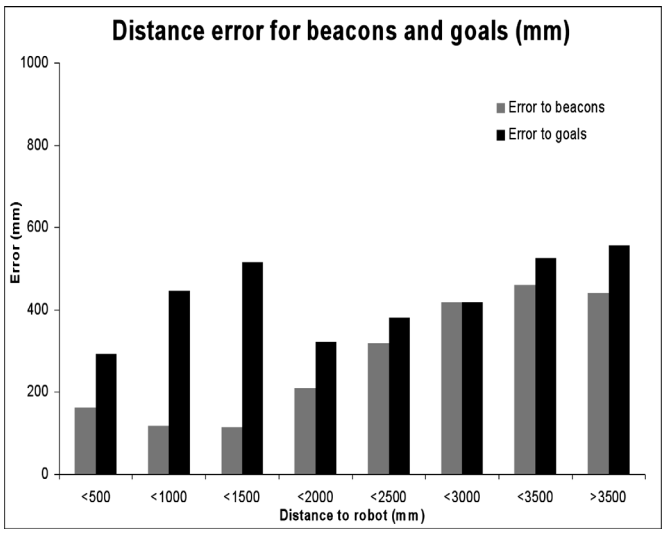


Fig. 2. Distance errors in the observation for beacons and goals.

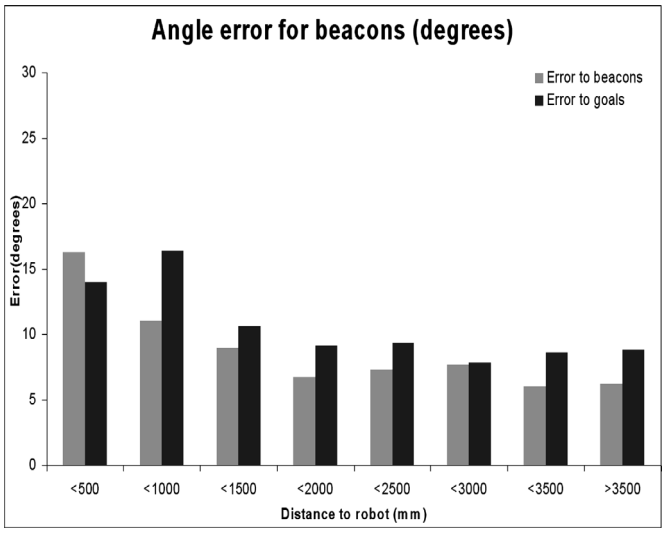


Fig. 3. Angle errors in the observation for beacons and goals.

Figure 2 presents a comparison of distance errors in observations for both beacons and goals. As can be seen, distance errors in beacons are smaller than ones for goals. Additionally, goal sensing has an increment in error if they are closer or farther than 2500 mm. Meanwhile beacon error is proportional to distance to the robot.

As another example, Fig. 3 shows angle observation errors for beacons and goals respectively. Thus, angle observation error in beacons is generally less than that in goals and these errors increase when a robot is closer.

3. Theoretical Background

The localization process for an Aibo robot in the RoboCup domain has already achieved much progress. In this section we show an analysis for three popular localization methods such as FM, EKF, and FM–EKF. As in previous implementations[9,6] the FM method offers an acceptable performance and the EKF offers more research acceptance and maturation. However, a hybrid perspective of FM–EKF has a combined advantage from these two independent methods.

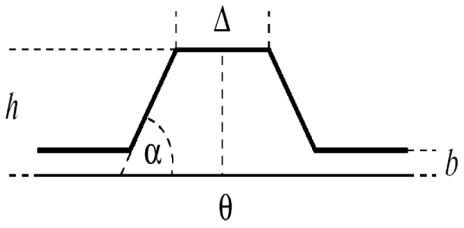


Fig. 4. Representation of the robot pose in a *fcell*.

A. Fuzzy Markov method

We use an FM grid-based method,[29] where a grid G_t contains a number of cells, and each grid element $G_t(x, y)$ holds a value of probability of a possible robot position in a range of [0, 1]. A fuzzy cell (*fcell*) is represented as a fuzzy trapezoid (Fig. 4) defined by a tuple $< \theta, \Delta, \alpha, h, b >$, where θ is robot orientation at the trapezoid centre with values in range of $[0, 2\pi]$; Δ is uncertainty in a robot orientation θ; h corresponds to *fcell* with a range of [0, 1]; α is the trapezoid slope and b is a correcting bias.

One of the most important aspects of a localization process based on Bayes filtering techniques is a phase cycle of prediction-observation-updating. If we look at such a phase cycle more closely, the *prediction* step adjusts movement information, meanwhile the *observation* step processes sensory information. Then, the *updating* step merges results from *prediction* and *observation* steps to obtain a new fuzzy grid map. Therefore, each step of the process sequence is described below:

(1) **Prediction step**. During this step, robot movements along grids are represented by a distribution which is continuously blurred. As is described in previous work,[9] blurring is based on odometry information reducing occupancy for robot movement (i.e., Fig. 5(c)). Thus, the grid state G'_t is obtained by performing a translation and rotation of G_{t-1} state distribution according to motion $\vec{m}$. Subsequently, this odometry-based blurring, B_t, is uniformly done for incorporating uncertainty to a motion state.

$$G'_t = f(G_{t-1}, \vec{m}) \otimes B_t \tag{9}$$

(2) **Observation step**. In this step, each observed landmark i is represented as a vector $\vec{r}_i$, which includes both range and bearing information obtained by the visual module. For each $\vec{r}_i$, a grid map $S_{i,t}$ is built such that $S_{i,t}(x, y, \theta|\vec{r}_i)$ corresponds to robot position at (x, y, θ) given an observation $\vec{r}$ at time t.

(3) **Updating step**. At this step, grid state G'_t obtained from the prediction step is merged with each state $S_{t,i}$ from the observation step. Afterwards, fuzzy intersection is done using a product operator as follows:

$$G_t = G'_t \times S_{t,1} \times S_{t,2} \times \cdots \times S_{t,n} \tag{10}$$

A simulated example of this process is shown in Fig. 5. The system starts with absolute uncertainty of robot pose, i.e., Fig. 5(a). Thereafter, the robot incorporates landmark and goal information where each grid state G_t is updated

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

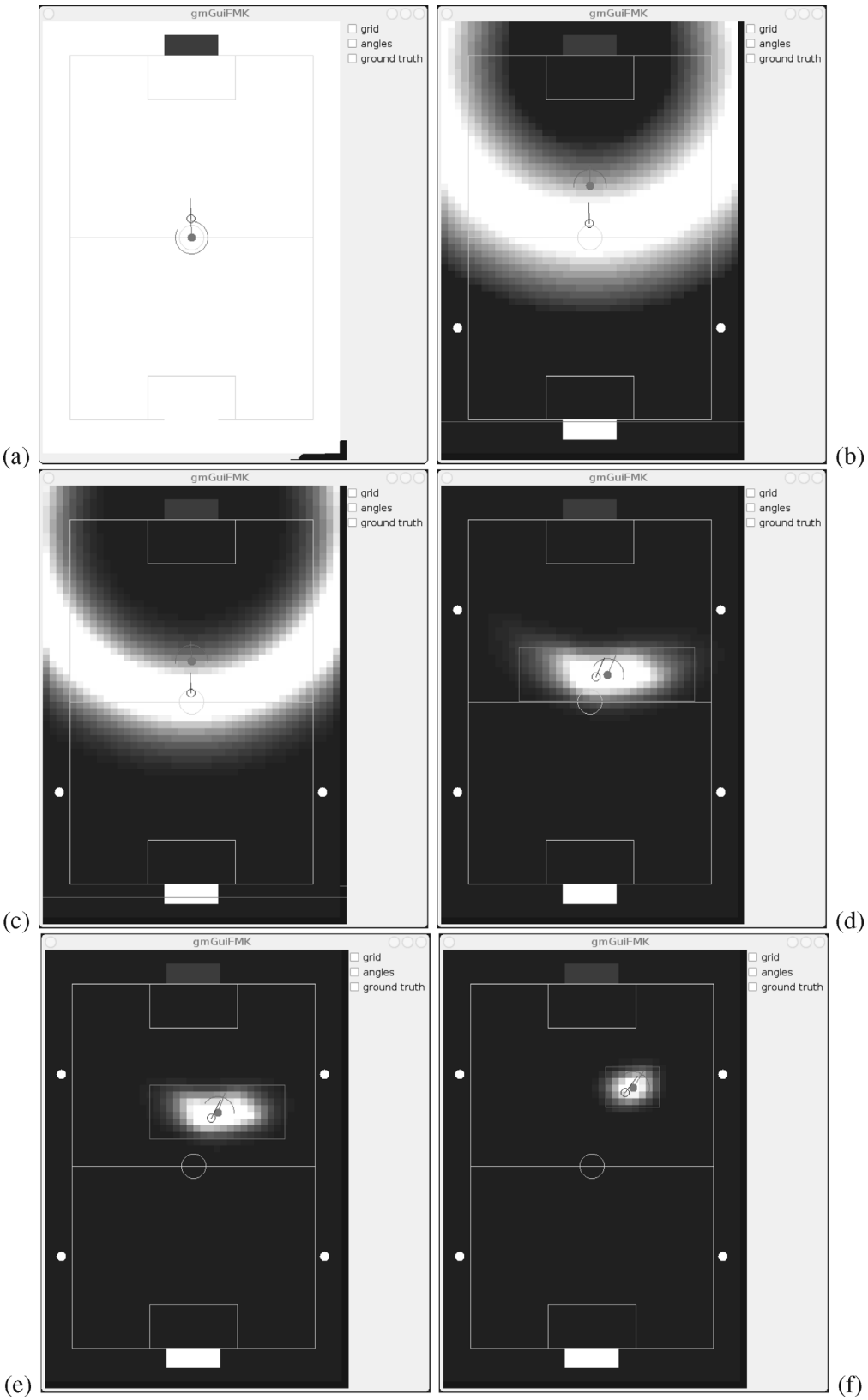


Fig. 5. This figure represents uncertainty in white colour from an absolute initialization (a) to a reduced area of robot pose in (f).

whenever an observation succeed, as is shown in Fig. 5(b). Subsequently, movements and observations of various landmarks enable the robot to localise, as shown in Fig. 5(c)–Fig. 5(f).

Since the performance of this method is judged by accuracy and computational cost, we selected a reasonable[9] *fcell* size of 20cm to balance two factors, i.e., for achieving good accuracy and less computing cost.

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

This localization method has the following advantages:

- Fast recovery from previous errors in the robot pose estimation.
- Fast recovery from kidnappings.
- Multi-hypothesis for robot pose (x, y).
- Much faster than classical Markovian approaches.

However its disadvantages, which we want to improve, are:

- Mono-hypothesis for orientation estimation.
- Very sensitive to sensor errors.
- False positives which make the method unstable in noisy conditions.
- Computational time increases dramatically.

B. Extended Kalman filter method

Techniques related to EKF have become one of the most popular tools for state estimation in robotics. Such approach contemplates a state vector for robot positioning which is related to environment perception and robot odometry. In this case, robot position is conformed by a vector s_t, consisting of (x, y, θ) in which (x, y) is robot position and θ for orientation.

$$\mathbf{s} = \begin{pmatrix} x_{robot} \\ y_{robot} \\ \theta_{robot} \end{pmatrix} \tag{11}$$

As a Bayesian filtering based method, EKF has *prediction* and *update* steps, described in detail below:

(1) **Prediction step**. This phase uses robot odometry information as input for filtering the state vector. Current robot state s_t^- is affected by robot odometry information and its noise estimation in an error estimation P_t^-. Initially, robot state is represented in:

$$s_t^- = f(s_{t-1}, u_t, w_t) \tag{12}$$

where the nonlinear function f relates the previous state s_{t-1}, control input $u_{t-1,}$ and the process noise w_t. Afterward, a covariance matrix P_t^- is used for representing errors in state estimation obtained at the previous step P_{t-1}. Therefore, the covariance matrix is related with the previous robot state and process noise, i.e.,

$$P_t^- = A_t P_{t-1} A_t^T + W_t Q_{t-1} W_t^T \tag{13}$$

where $A_t P_{t-1} A_t^T$ is a progression of P_{t-1} along a new movement and A_t is defined as follows:

$$A_t = \frac{\partial f}{\partial s} = \begin{pmatrix} 1 & 0 & -u_t^{lat} cos\theta_t - u_t^{lin} sen\theta_{t-1} \\ 0 & 1 & u_t^{lin} cos\theta_t - u_t^{lat} sen\theta_{t-1} \\ 0 & 0 & 1 \end{pmatrix} \tag{14}$$

and $W_t Q_{t-1} W_t^T$ represents odometry noise, W_t is a Jacobian for motion state approximation and Q_t is a covariance matrix as follows:

$$Q_t = E\left[w_t w_t^T\right] \tag{15}$$

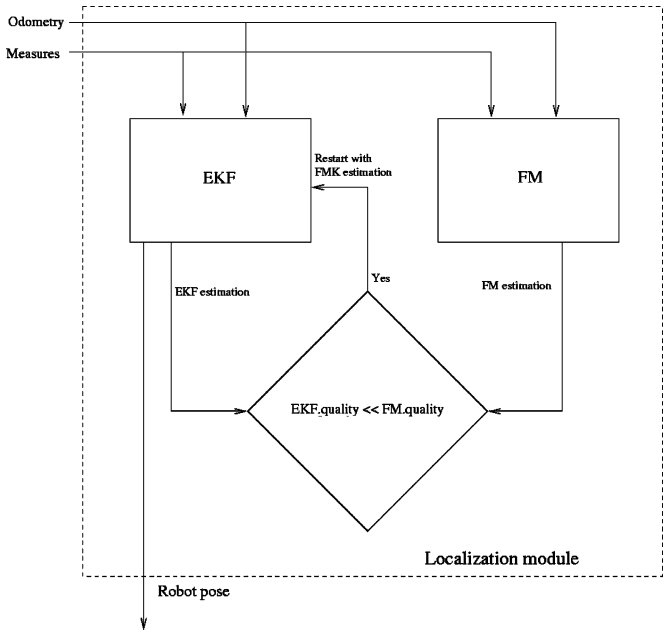


Fig. 6. Hybrid localization process.

Note that the Sony AIBO robot may not be able to observe a beacon at each step. Therefore, the frequency of odometry calculation is higher than visual sensor measurements. For this reason, this step is executed independently as follows:[11]

$$s_t = s_t^- \tag{16}$$

$$P_t = P_t^- \tag{17}$$

(2) **Updating step**. At this step, sensor data and noise covariance P_t are used for obtaining a new state vector. Besides, the sensor model is updated using measured landmarks $m_{1\cdots6,(x,y)}$. Thus, each z_t^i of i landmarks requires a vector $(r_t^i, \phi t^i)$. In order to obtain an updated state, we use:

$$s_t = s_{t-1} + K_t^i\left(z_t^i - \hat{z}_t^i\right) = s_{t-1} + K_t^i\left(z_t^i - h^i(s_{t-1})\right) \tag{18}$$

where $h^i(s_{t-1})$ is a predicted measurement calculated from the following non-linear functions:

$$\hat{z}_t^i = h^i(s_{t-1}) = \begin{pmatrix} \sqrt{\left(m_{t,x}^i - s_{t-1,x}\right)^2 + \left(m_{t,y}^i - s_{t-1,y}\right)} \\ atan2\left(m_{t,x}^i - s_{t-1,x}, m_{t,y}^i - s_{t-1,y}\right) - s_{t-1,\theta} \end{pmatrix} \tag{19}$$

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

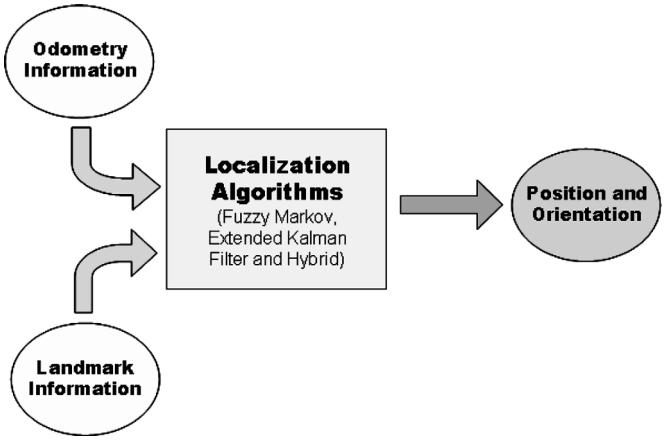


Fig. 7. Information flow for localization.

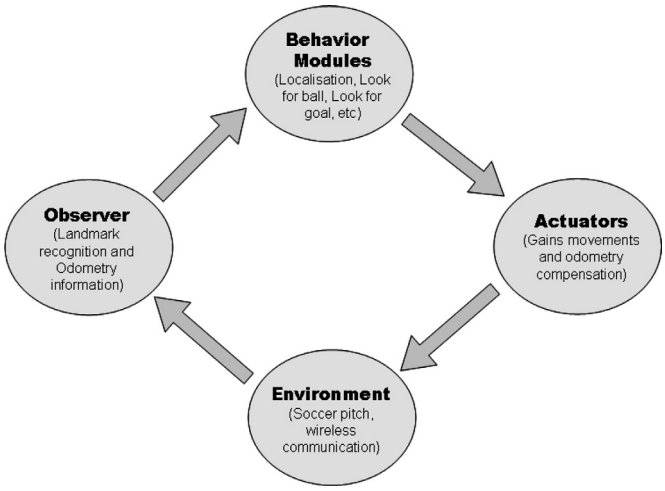


Fig. 8. System overview.

The *Kalman gain*, K_t^i, is obtained from the next equation:

$$K_t^i = P_{t-1}\left(H_t^i\right)^T\left(S_t^i\right)^{-1} \tag{20}$$

where S_t^i is uncertainty for each predicted measurement $\hat{z}_t^i$ and is calculated as follows:

$$S_t^i = H_t^i P_{t-1}\left(H_t^i\right)^T + R_t^i \tag{21}$$

and H_t^i describes how robot position is changing as follows:

$$H_t^i = \frac{\partial h^i(s_{t-1})}{\partial s_t} \tag{22}$$

$$= \begin{pmatrix} -\dfrac{m_{t,x}^i - s_{t-1,x}}{\sqrt{\left(m_{t,x}^i - s_{t-1,x}\right)^2 + \left(m_{t,y}^i - s_{t-1,y}\right)^2}} & -\dfrac{m_{t,y}^i - s_{t-1,y}}{\sqrt{\left(m_{t,x}^i - s_{t-1,x}\right)^2 + \left(m_{t,y}^i - s_{t-1,y}\right)^2}} & 0 \\ \dfrac{m_{t,y}^i - s_{t-1,y}}{\left(m_{t,x}^i - s_{t-1,x}\right)^2 + \left(m_{t,y}^i - s_{t-1,y}\right)^2} & -\dfrac{m_{t,x}^i - s_{t-1,x}}{\left(m_{t,x}^i - s_{t-1,x}\right)^2 + \left(m_{t,y}^i - s_{t-1,y}\right)^2} & -1 \\ 0 & 0 & 0 \end{pmatrix}$$

where R_t^i is measurement noise which was empirically obtained and P_t is calculated using the following equation:

$$P_t = \left(I - K_t^i H_t^i\right)P_{t-1} \tag{23}$$

As not all $\hat{z}_t^i$ values are obtained for every observation, z_t^i are correct references for each observed value and δ_t^i is a likelihood measurement obtained from Eq. (24) with a threshold value between 5 and 100, which varies based on localization quality.

$$\delta_t^i = \left(z_t^i - \hat{z}_t^i\right)^T\left(S_t^i\right)^{-1}\left(z_t^i - \hat{z}_t^i\right) \tag{24}$$

C. FM–EKF method

We propose to merge FM and EKF algorithms in order to achieve computational efficiency, robustness, and reliability. In particular, the integrated FM–EKF is able to deal with noisy visual information and inaccurate odometry data in order to obtain the most reliable position from both approaches.

The hybrid procedure is fully described in Algorithm 1. Initially, the hybrid algorithm conditions require the FM algorithm to implement a *fcell* grid size of (500–1000 mm) initialised in the field centre. Subsequently, a variable is iterated for controlling FM results. This variable is used for measuring robot positioning after being compared with the quality of EKF estimates. Then, localization quality indicates whenever EKF can be resetted in case the robot is lost or EKF position is out of FM range. Eventually, each algorithm step is realised depending on the execution of motion and perception models. Consequently, the localization process follows a predict-correct scheme.

```
Initialise position_FM over all pitch;
Initialise position_EKF over all pitch;
while true do
    Predict position_FM using motion model;
    Predict position_EKF using motion model;
    Correct position_FM using perception model;
    Correct position_EKF using perception model;
    if (quality(position_FM) ≫ quality(position_EKF) then
        Initialise position_EKF to position_FM
    end
    robot position ← position_EKF;
end
```

Algorithm 1: FM and EKF combined methods algorithm.

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

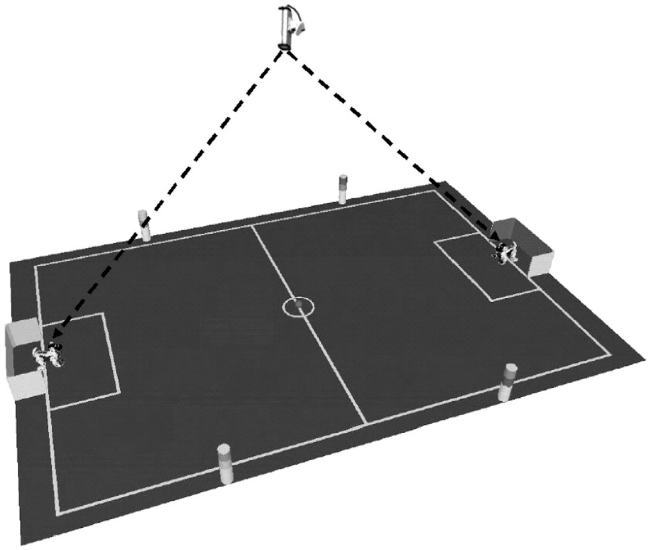


Fig. 9. The robot environment: a robot overhead tracking system, a soccer pitch with natural (pitch lines) and artificial landmarks (coloured beacons and goals), and two Aibo walking robots.

The hybrid approach implements both FM and EKF methods for improving robot pose estimation. Even though they are processed independently, they use the same sensor and motion information to maintain independent robot pose estimation. Note that the FM localization is a robust solution for noisy information, but computational expensive and cannot operate in real time. Also, its accuracy is inversely proportional to the *fcell* size. In contrast, EKF is an efficient and accurate positioning method. Therefore, the hybrid method combines FM grid accuracy with EKF tracking efficiency, as shown in Fig. 6. The localization information flow is described in Fig. 7. Information from sensors (odometry and feature extraction system) is used by localization algorithms for obtaining robot positioning and orientation information.

4. System Overview

Our system consists of three asynchronously coordinated parts: an overhead tracking system, an Aibo legged robot, and human-robot control software. What is more, robot software contains an observer and a number of behavior modules as shown in Fig. 8. The observer is a perception module to collect environment information for navigation. It is organised as a set of possible states realised by a robot to interact within its environment. Robot behavior acts as a decision maker and is executed independently from the observer module with the same processing resources and information.

The legged robot used in our research is a Sony Aibo ERS-7 robot that has an embedded MIPS processor running at 576MHz, 64MB memory, a 350K-pixel colour camera, a pair of infrared sensors, and a LAN wireless IEEE 8011b card. The robot's environment used in our experiments is a football pitch used in the RoboCup four legged league competition. The pitch contains four visual landmarks and two goals with distinctive colours, as well as white field lines, as shown in Fig. 9. Note that all the landmarks and soccer environment follow the 2006 RoboCup four Legged League rules.

An overhead tracking camera was used in our experiment in order to provide a ground truth for robot tracking movements. The overhead tracking system has a mean error

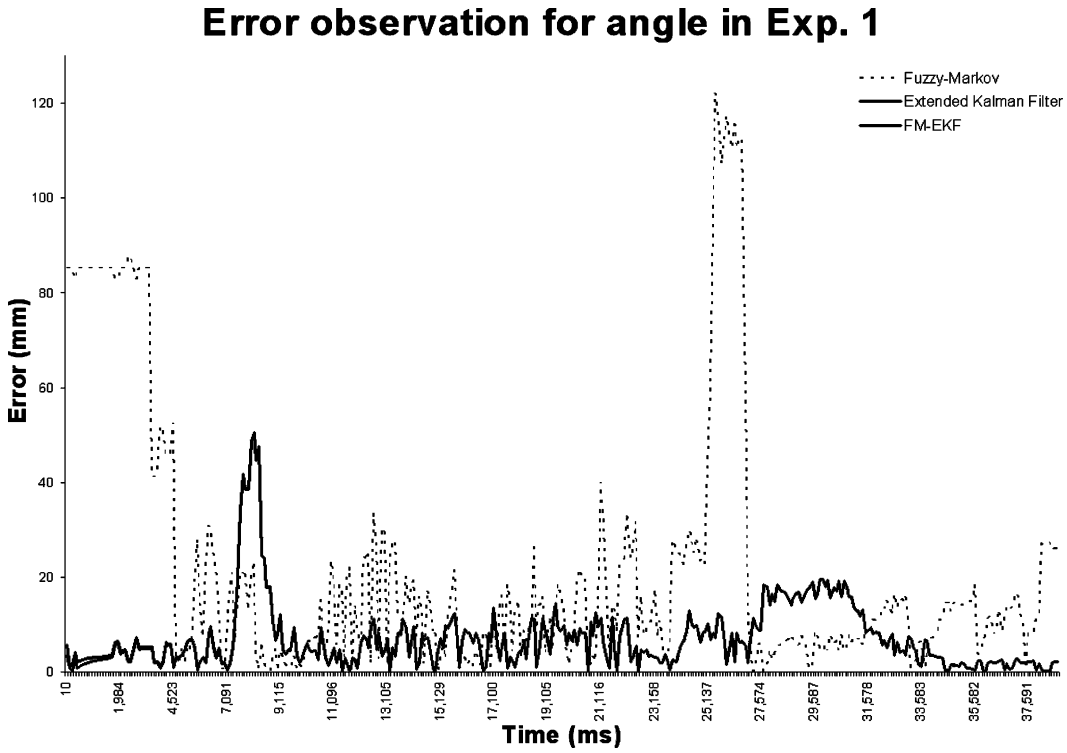


Fig. 10. Orientation errors during a walk along the pitch.

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

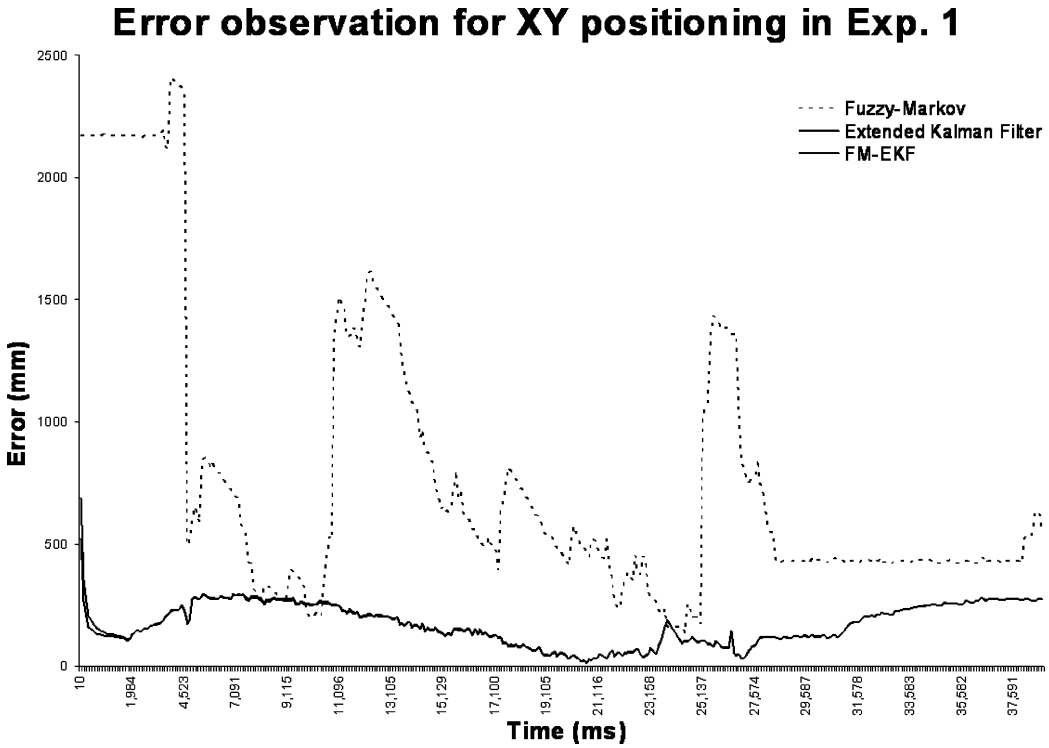


Fig. 11. Positioning errors during a walk along the pitch.

of (72.4 mm, 60.0 mm) and an average standard deviation of (63.5 mm, 46.7 mm) for x and y coordinates.

5. Experimental Results

The experiments were carried out in three stages of work: (i) simple movements; (ii) combined behaviors; and (iii) kidnapped robot. Each experiment set is to show robot positioning abilities in a RoboCup environment. The total set of experiment updates were 15, with 14123 updates in total. In each experimental set, the robot poses estimated by EKF, FM and FM–EKF localization methods are compared with the ground truth generated by the overhead vision system. In addition, each experiment set is compared respectively within its processing time. Experimental sets are described below:

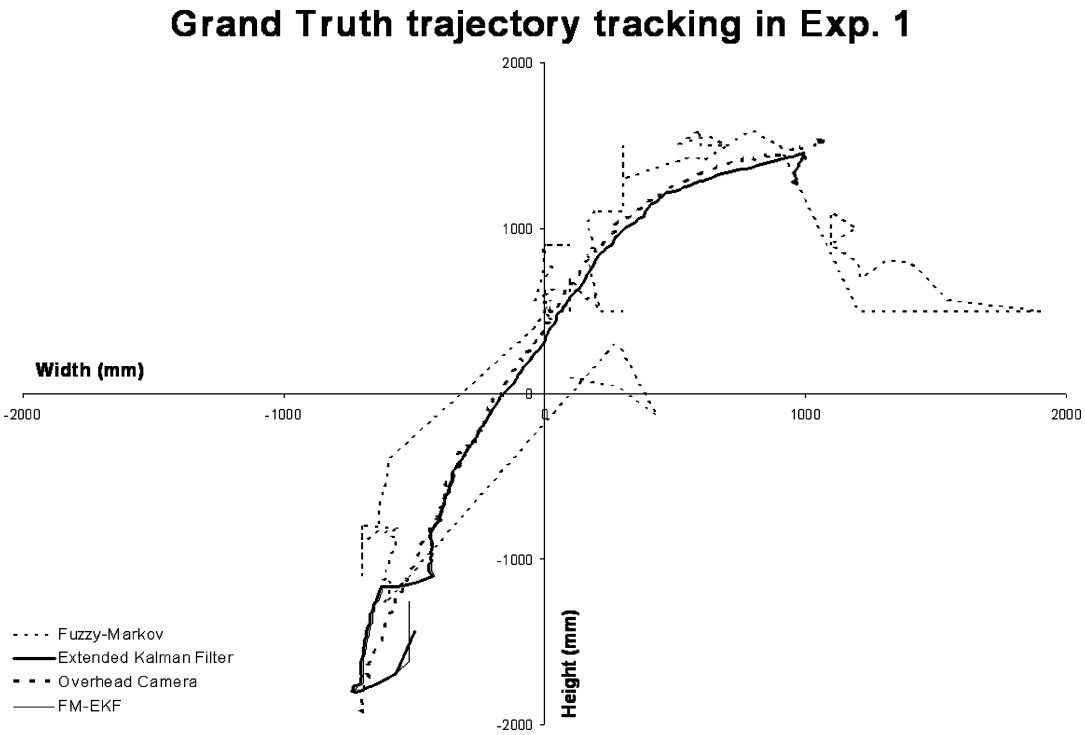


Fig. 12. Robot trajectories estimated by EKF, FM, FM–EKF and the overhead camera.

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

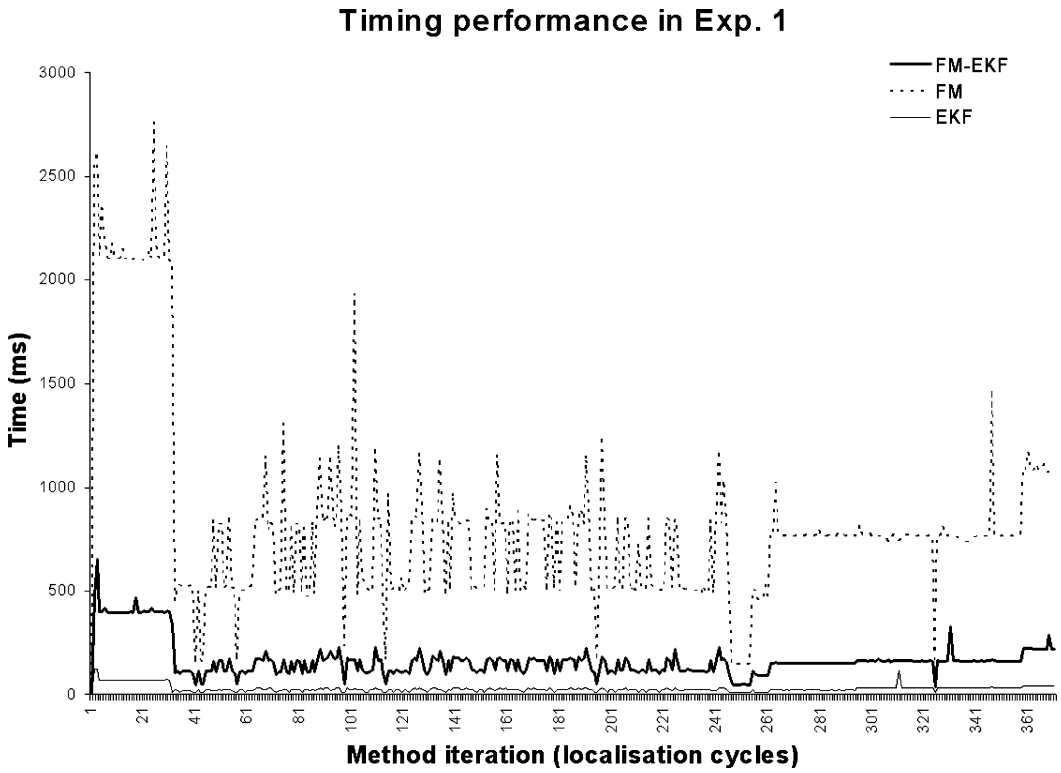


Fig. 13. Processing time per algorithm iteration during a walk along the pitch.

(1) **Simple Movements**. This stage includes straight and circular robot trajectories in forward and backward directions within the pitch.
(2) **Combined Behavior**. This stage is composed by a pair of high level behaviors. Our first experiment consists of reaching a predefined group of co-ordinated points along the pitch. Then, the second experiment is about playing alone and with another dog to obtain localization results during a long period.

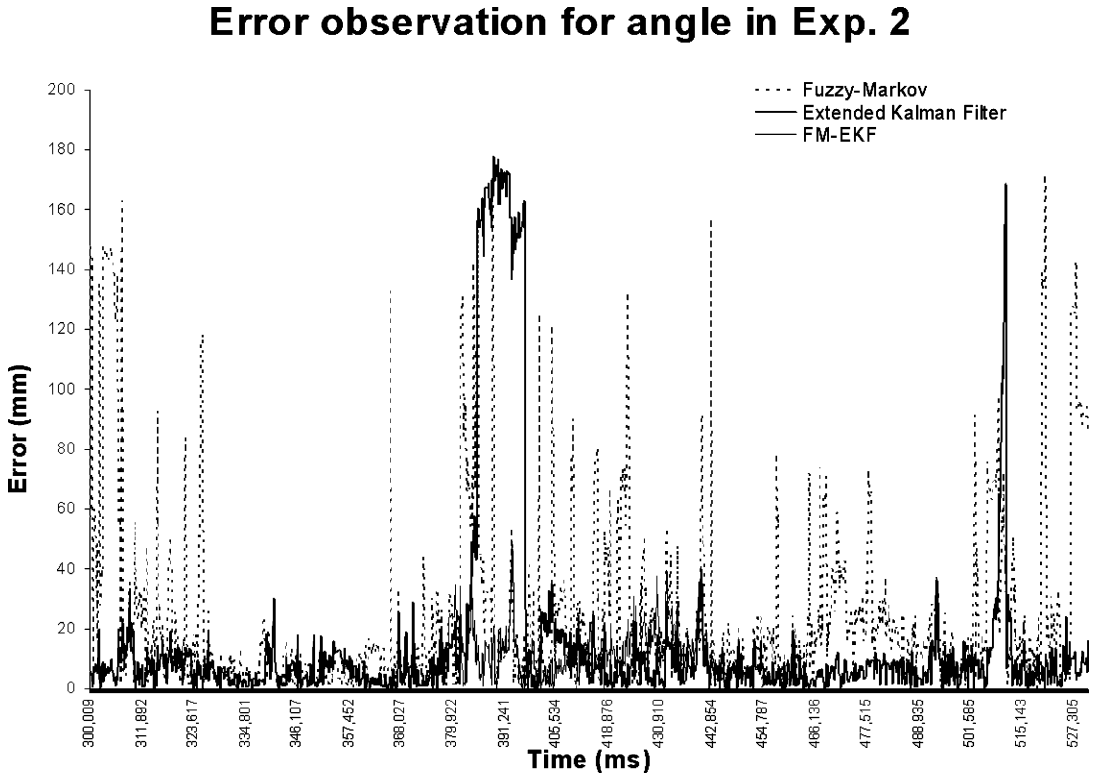


Fig. 14. Orientation errors during reaching predefined points.

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

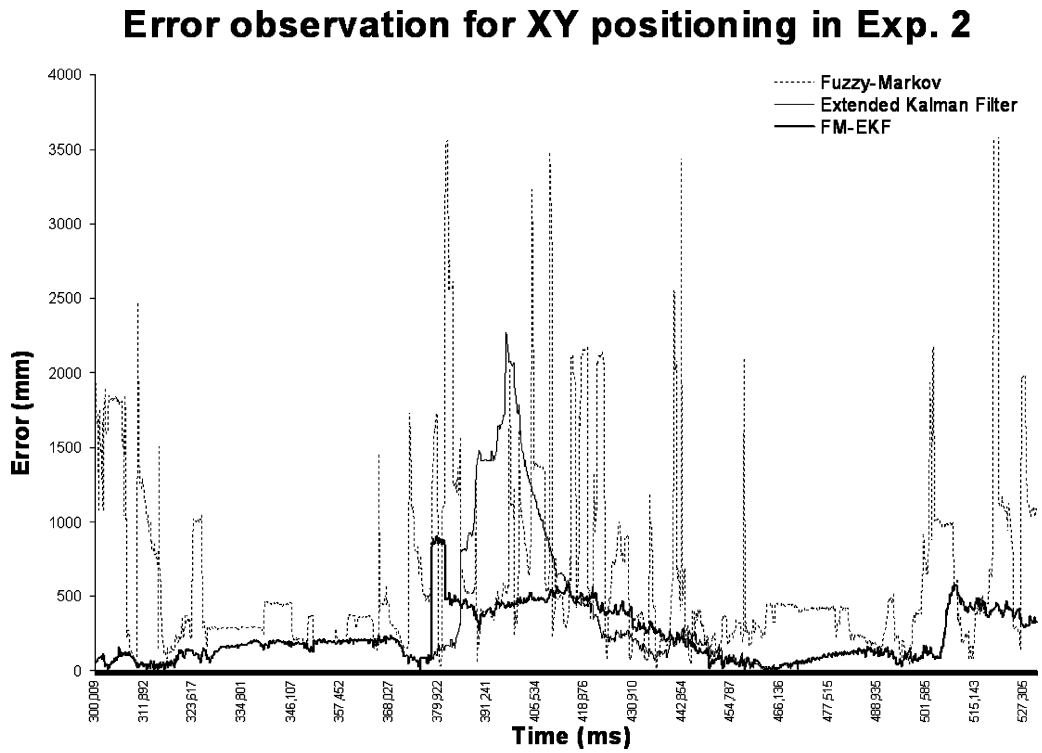


Fig. 15. Positioning errors during reaching predefined points.

(3) **Kidnapped Robot**. This stage is realised randomly in sequences of kidnap time and pose. In each kidnap the objective is to obtain information about where the robot is and how fast it can localise again.

In the Simple Movement experiment set, the robot follows a trajectory in order to obtain a variety of visible points along the pitch. Figures 10 and 11 show angle and position errors respectively. Figure 12 presents the trajectories estimated by EKF, FM, FM–EKF methods and the overhead vision system. Figure 13 shows the processing time for EKF, FM, FM–EKF methods, in which the proposed FM–EKF method is faster than the FM method, but slower than the EKF method. As can be seen, during this experiment high peaks in angle or

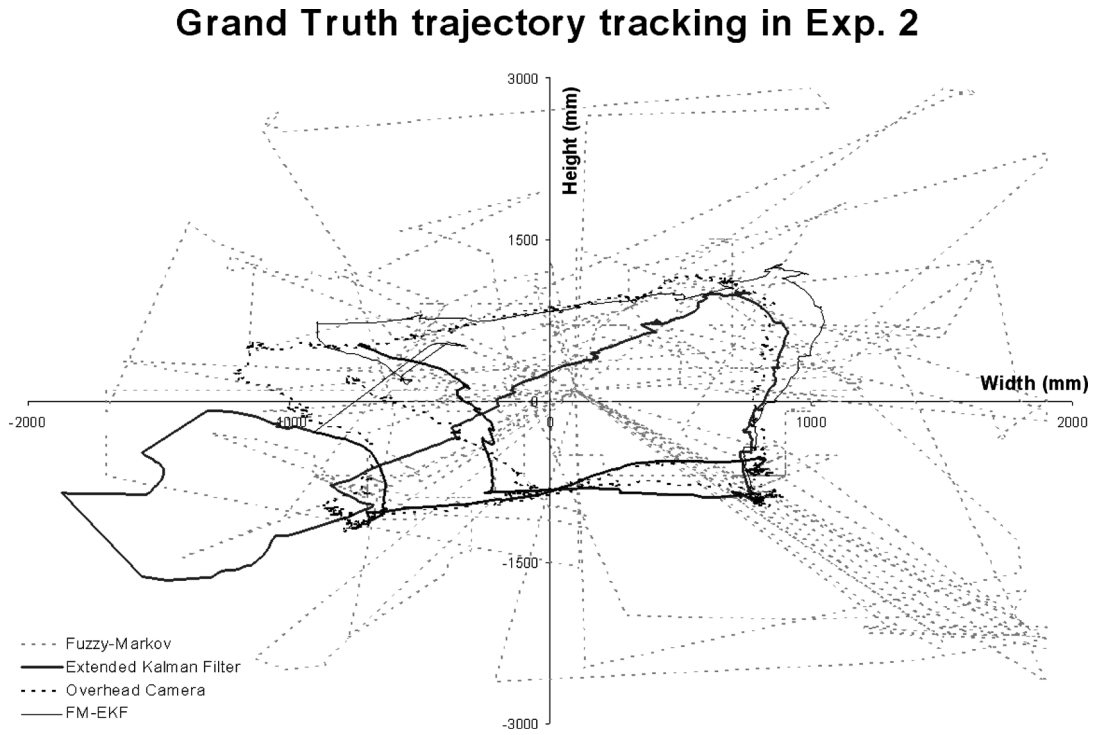


Fig. 16. Robot trajectories estimated by EKF, FM, FM–EKF and the overhead camera as a Grand Truth system.

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

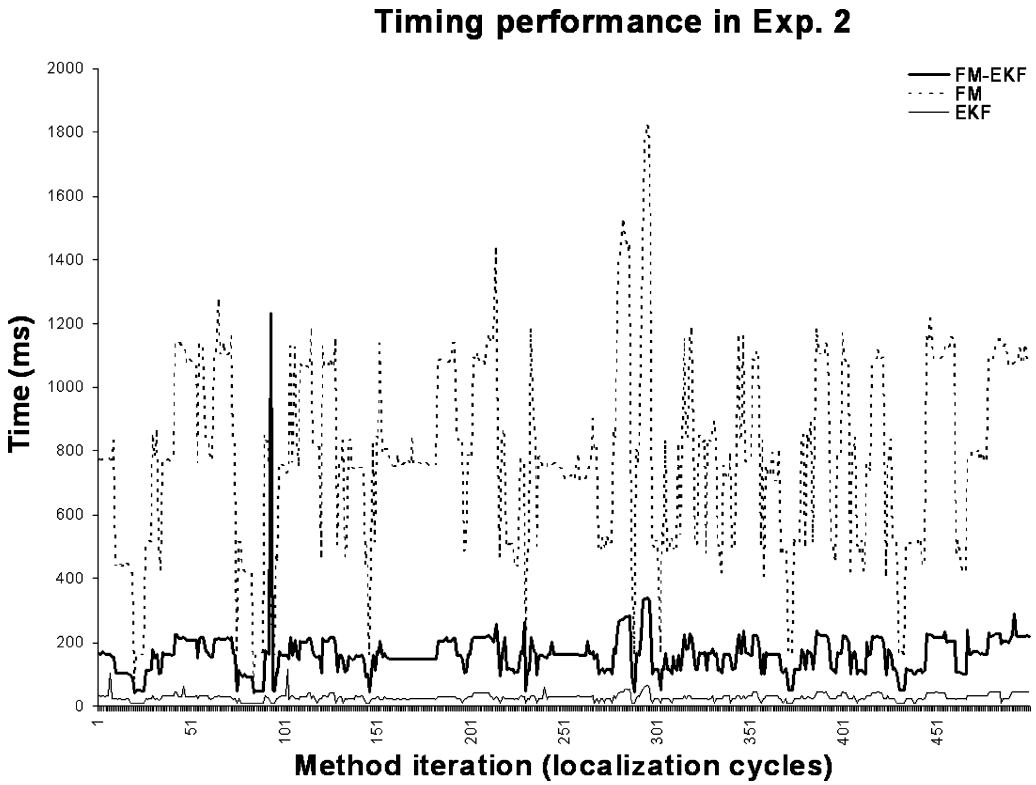


Fig. 17. Processing time per algorithm iteration during reaching predefined points.

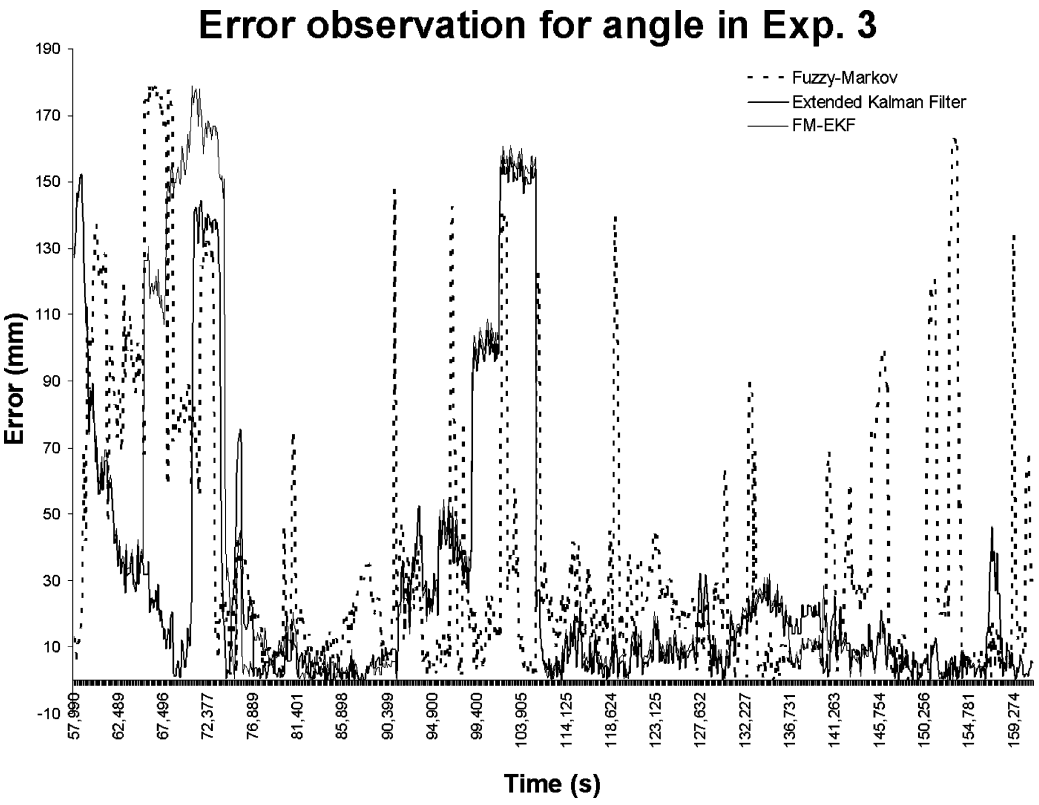


Fig. 18. Orientation errors during a kidnapping sequence.

distance error occur whenever position is different from that reported by the overhead vision system. This is caused by a lack of landmark perception.

For the Combined Behavior experiment we evaluate two dissimilar experiments: (a) robot reaching predefined points as accurately as possible and (b) a playing session for a robot to play a session alone and against an opponent. Figures 14 and 15 present orientation and position errors whenever the robot reaches predefined points. Figure 16 show the difference of robot trajectories estimated by FM–EKF and

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

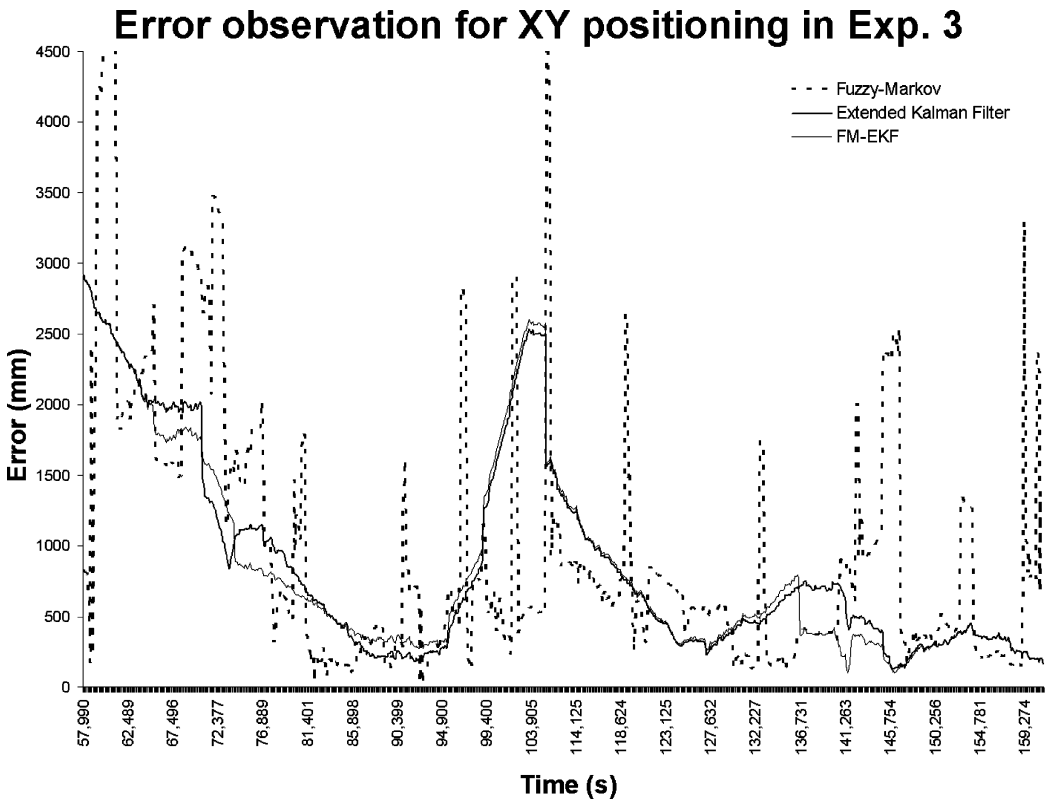


Fig. 19. Positioning errors during a kidnapping sequence.

overhead vision system. Figure 17 shows the processing time per algorithm iteration for EKF, FM, FM–EKF during the robot reaching some predefined points.

Similar to previous experiments, periods of landmark absence causes increasing error. From the performance perspective, EKF is the fastest localization method, FM–EKF is in the middle, and FM is the slowest one.

In the last experiment, the robot was randomly kidnapped in terms of time, position, and orientation. After the robot is manually deposited in a different pitch zone, its localization

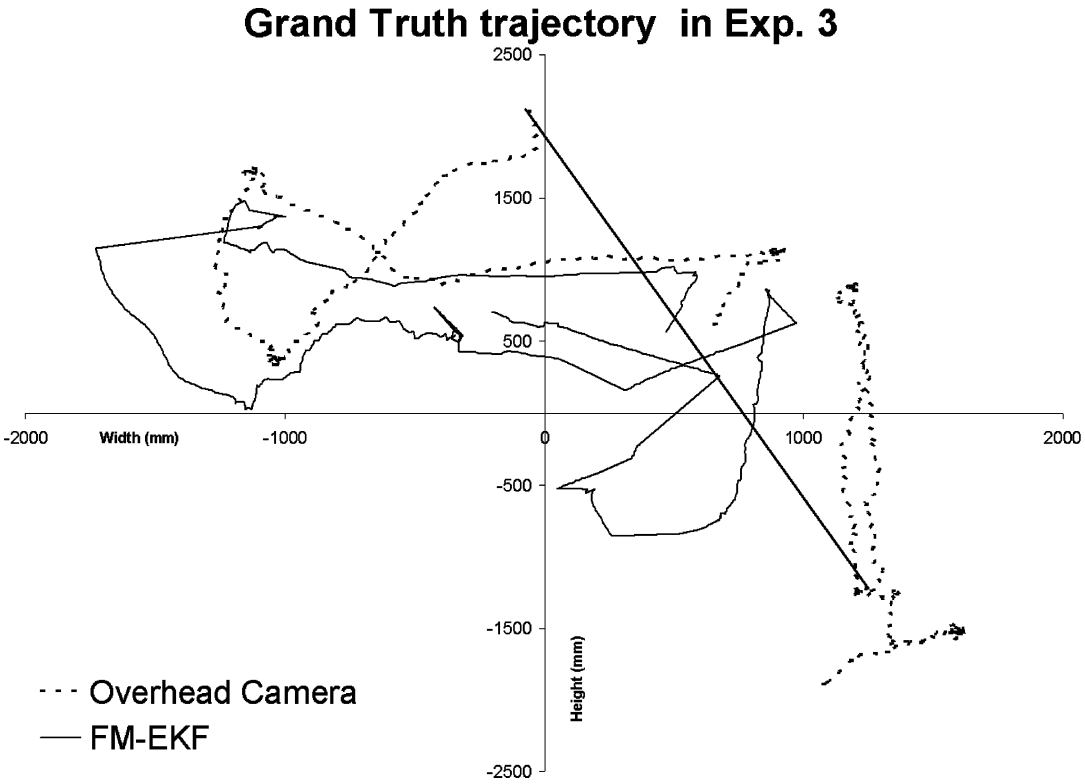


Fig. 20. Robot trajectories estimated by FM–EKF and the overhead camera.

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

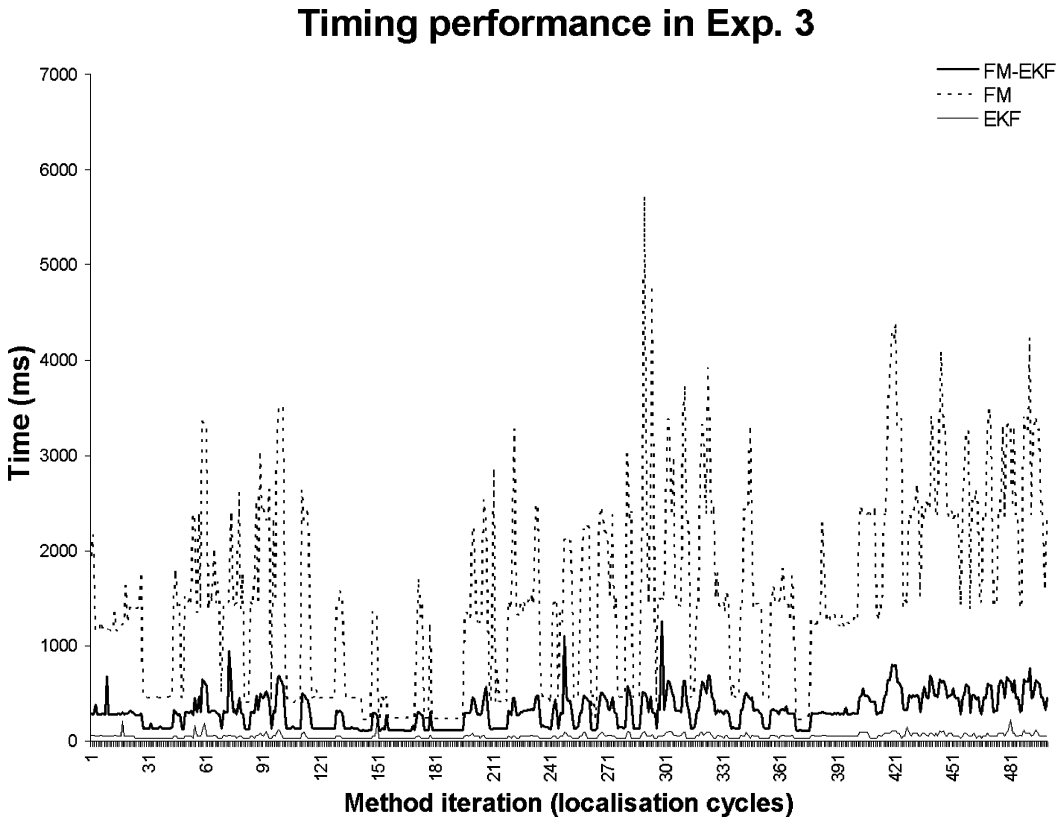


Fig. 21. Processing time per algorithm iteration during a kidnapping sequence.

performance is evaluated. Figures 18 and 19 show orientation and position errors during a kidnapping sequence. Figure 20 presents trajectories estimated by FM–EKF and overhead vision system. Figure 21 shows the processing time per iteration for three algorithms during reaching of predefined points. Results from kidnapped experiments show resetting transition from a local minimum to fast convergence within 3.23 s. Even EKF has the most efficient computation time, FM–EKF offers the most stable performance and is a most suitable method for a dynamic indoor environment.

6. Conclusions

This paper presents a novel hybrid approach to fast and accurate robot localization using an active vision platform. The proposed FM–EKF method integrates FM and EKF algorithms using the visual and odometry information from a legged robot. Experimental results show that the hybrid method offers a more stable performance and good localization accuracy for a legged robot which has noisy odometry information. Further research will focus on the reinforcement of the quality in observer mechanisms for odometry and visual perception, as well as the improvement of landmark recognition performance.

Acknowledgements

We would like to thank TeamChaos (http://www.teamchaos.es) for facilities and capabilities of their programming work and technical team from Department of Computing and Electronic Systems at Essex University (http://essexrobotics.essex.ac.uk), for their support and help in this research. Part of this research was also supported by a CONACyT (Mexican government) scholarship with reference number 178622 and by Spanish government grants *DPI2004-07993-C03-01, DPI2004-07993-C03-02*, and *S-0505/DPI/0176* corresponding to *Ditpa* and *Robocity 2030* projects respectively.

References

1. H. Baltzakis and P. Trahanias, "Hybrid Mobile Robot Localization Using Switching State-Space Models," *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, USA (2002) pp. 366–373.
2. P. Buschka, A. Saffiotti and Z. Wasik, "Fuzzy Landmark-Based Localization for a Legged Robot," *Proceedings of IEEE Intelligent Robots and Systems*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kagawa University, Takamatsu, Japan, (2000) pp. 1205–1210.
3. T. Duckett and U. Nehmzow, "Performance Comparison of Landmark Recognition Systems for Navigating Mobile Robots," *Proceedings of 17th National Conference on Artificial Intelligence (AAAI-2000)*, AAAI Press/The MIT Press (2000) pp. 826–831.
4. D. Fox, W. Burgard, F. Dellaert and S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," *Proceedings of AAAI/IAAI*, Orlando, Florida, USA, (1999), pp. 343–349.
5. D. Fox, W. Burgard and S. Thrun, "Markov Localization for Reliable Robot Navigation and People Detection," In Sensor Based Intelligent Robots, (H. I. Christens *et al.*, ed), Dagstuhl Castle, Germany, (1998), pp. 1–20.
6. J.-S. Gutmann, "Markov–Kalman Localization for Mobile Robots," *International Conference on Pattern Recognition (2)*, Quebec, Canada, (2002), pp. 601–604.
7. J.-S. Gutmann, W. Burgard, D. Fox and K. Konolige, "An Experimental Comparison of Localization Methods," *Proceedings of IEEE/RSJ International Conference on Intelligen Robots and Systems* (Oct. 13–17, 1998) vol. 2, pp. 736–743.

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press

8. K. Hatice, B. Celik and H. L. Akin, "Comparison of localization methods for a robot soccer team," *Int. J. Adv. Robot. Syst.* **3**(4), 295–302 (2006).
9. D. Herrero-Pérez, H. M. Artínez Barberá and A. Saffiotti, "Fuzzy Self-Localization Using Natural Features in the Four-Legged League," In *RoboCup 2004: Robot Soccer World Cup VIII* (D. Nardi, M. Riedmiller and C. Sammut, eds.) (LNAI. Springer-Verlag, Berlin, DE, 2004). Online at http://www.aass.oru.se/˜asaffio/. pp. 110–121.
10. P. Jensfelt, D. Austin, O. Wijk and M. Andersson, "Feature Based Condensation for Mobile Robot Localization," *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, USA, (2000) pp. 2531–2537.
11. E. Kiriy and M. Buehler, "Three-state extended kalman filter for mobile robot localization," Tech. Rep. TR-CIM 05.07, McGill University, Montreal, Canada, April 2002.
12. S. Kristensen and P. Jensfelt, "An Experimental Comparison of Localisation Methods," *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Navada, USA, (Oct. 27–31, 2003) pp. 992–997.
13. K. Tanaka, Y. Kimuro, N. Okada and E. Kondo, "Global Localization with Detection of Changes in Non-Stationary Environments," *Proceedings of IEEE International Conference on Robotics and Automation* (Apr. 26–May 1, 2004) pp. 1487–1492.
14. S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz "Probabilistic algorithms and the interactive museum tour-guide robot Minerva," *Int. J. Robot. Res.* **19**(11), 972–999 (2000).
15. S. Thrun, D. Fox, W. Burgard and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *J. Artificial Intell.* **128**(1–2), 99–141 (2001).

https://doi.org/10.1017/S0263574708004414 Published online by Cambridge University Press