1

Visual Localisation of quadruped walking robots

Renato Samperio and Huosheng Hu
School of Computer Science and Electronic Engineering, University of Essex
United Kingdom

1. Introduction

Recently, several solutions to the robot localisation problem have been proposed in the scientific community. In this chapter we present a localisation of a visual guided quadruped walking robot in a dynamic environment. We investigate the quality of robot localisation and landmark detection, in which robots perform the RoboCup competition (Kitano et al., 1997). The first part presents an algorithm to determine any entity of interest in a global reference frame, where the robot needs to locate landmarks within its surroundings. In the second part, a fast and hybrid localisation method is deployed to explore the characteristics of the proposed localisation method such as processing time, convergence and accuracy.
In general, visual localisation of legged robots can be achieved by using artificial and natural landmarks. The landmark modelling problem has been already investigated by using predefined landmark matching and real-time landmark learning strategies as in (Samperio & Hu, 2010). Also, by following the pre-attentive and attentive stages of previously presented work of (Quoc et al., 2004), we propose a landmark model for describing the environment with "interesting" features as in (Luke et al., 2005), and to measure the quality of landmark description and selection over time as shown in (Watman et al., 2004). Specifically, we implement visual detection and matching phases of a pre-defined landmark model as in (Hayet et al., 2002) and (Sung et al., 1999), and for real-time recognised landmarks in the frequency domain (Maosen et al., 2005) where they are addressed by a similarity evaluation process presented in (Yoon & Kweon, 2001). Furthermore, we have evaluated the performance of proposed localisation methods, Fuzzy-Markov (FM), Extended Kalman Filters (EKF) and an combined solution of Fuzzy-Markov-Kalman (FM-EKF),as in (Samperio et al., 2007)(Hatice et al., 2006).
The proposed hybrid method integrates a probabilistic multi-hypothesis and grid-based maps with EKF-based techniques. As it is presented in (Kristensen & Jensfelt, 2003) and (Gutmann et al., 1998), some methodologies require an extensive computation but offer a reliable positioning system. By cooperating a Markov-based method into the localisation process (Gutmann, 2002), EKF positioning can converge faster with an inaccurate grid observation. Also. Markov-based techniques and grid-based maps (Fox et al., 1998) are classic approaches to robot localisation but their computational cost is huge when the grid size in a map is small (Duckett & Nehmzow, 2000) and (Jensfelt et al., 2000) for a high resolution solution. Even the problem has been partially solved by the Monte Carlo (MCL) technique (Fox et al., 1999), (Thrun et al., 2000) and (Thrun et al., 2001), it still has difficulties handling environmental changes (Tanaka et al., 2004). In general, EKF maintains a continuous estimation of robot position, but can not manage multi-hypothesis estimations as in (Baltzakis & Trahanias, 2002).

Moreover, traditional EKF localisation techniques are computationally efficient, but they may also fail with quadruped walking robots present poor odometry, in situations such as leg slippage and loss of balance. Furthermore, we propose a hybrid localisation method to eliminate inconsistencies and fuse inaccurate odometry data with costless visual data. The proposed FM-EKF localisation algorithm makes use of a fuzzy cell to speed up convergence and to maintain an efficient localisation. Subsequently, the performance of the proposed method was tested in three experimental comparisons: (i) simple movements along the pitch, (ii) localising and playing combined behaviours and c) kidnapping the robot.
The rest of the chapter is organised as follows. Following the brief introduction of Section 1, Section 2 describes the proposed observer module as an updating process of a Bayesian localisation method. Also, robot motion and measurement models are presented in this section for real-time landmark detection. Section 3 investigates the proposed localisation methods. Section 4 presents the system architecture. Some experimental results on landmark modelling and localisation are presented in Section 5 to show the feasibility and performance of the proposed localisation methods. Finally, a brief conclusion is given in Section 6.

2. Observer design

This section describes a robot observer model for processing motion and measurement phases. These phases, also known as *Predict* and $Update$, involve a state estimation in a time sequence for robot localisation. Additionally, at each phase the state is updated by sensing information and modelling noise for each projected state.

2.1 Motion Model

The state-space process requires a state vector as processing and positioning units in an observer design problem. The state vector contains three variables for robot localisation, i.e., 2D position (x, y) and orientation (θ). Additionally, the prediction phase incorporates noise from robot odometry, as it is presented below:

$$\begin{pmatrix} x_t^- \\ y_t^- \\ \theta_t^- \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} (u_t^{lin} + w_t^{lin})cos\theta_{t-1} - (u_t^{lat} + w_t^{lat})sin\theta_{t-1} \\ (u_t^{lin} + w_t^{lin})sin\theta_{t-1} + (u_t^{lat} + w_t^{lat})cos\theta_{t-1} \\ u_t^{rot} + w_t^{rot} \end{pmatrix} \tag{4.9}$$

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 in errors of odometry. Also, $t-1$ refers to the time of the previous time step and t to the time of the current step.
In general, state estimation is a weighted combination of noisy states using both priori and posterior estimations. Likewise, assuming that v is the measurement noise and w is the process noise, the expected value of the measurement R and process noise Q covariance matrixes are expressed separately as in the following equations:

$$R = E[vv^t] \tag{4.10}$$

$$Q = E[ww^t] \tag{4.11}$$

The measurement noise in matrix R represents sensor errors and the Q matrix is also a confidence indicator for current prediction which increases or decreases state uncertainty. An odometry motion model, u_{t-1} is adopted as shown in Figure 1. Moreover, Table 1 describes all variables for three dimensional *(linear, lateral and rotational)* odometry information where $(\widehat{x}, \widehat{y})$ is the estimated values and (x, y) the actual states.

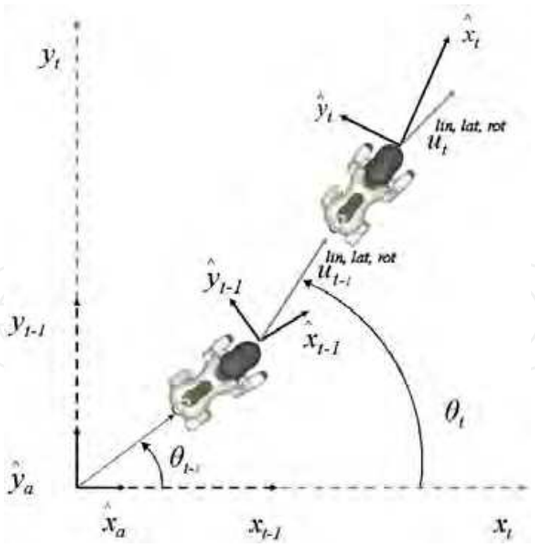


Fig. 1. The proposed motion model for Aibo walking robot

According to the empirical experimental data, the odometry system presents a deviation of 30% on average as shown in Equation (4.12). Therefore, by applying a transformation matrix W_t from Equation (4.13), noise can be addressed as robot uncertainty where θ points the robot heading.

$$Q_t = \begin{pmatrix} (0.3u_t^{lin})^2 & 0 & 0 \\ 0 & (0.3u_t^{lat})^2 & 0 \\ 0 & 0 & (0.3u_t^{rot} + \frac{\sqrt{(u_t^{lin})^2+(u_t^{lat})^2}}{500})^2 \end{pmatrix} \tag{4.12}$$

$$W_t = fw = \begin{pmatrix} cos\theta_{t-1} & -sen\theta_{t-1} & 0 \\ sen\theta_{t-1} & cos\theta_{t-1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.13}$$

2.2 Measurement Model

In order to relate the robot to its surroundings, we make use of a landmark representation. The landmarks in the robot environment require notational representation of a measured vector f_t^i for each i-th feature as it is described in the following equation:

$$f(z_t) = \{f_t^1, f_t^2, ...\} = \{\begin{pmatrix} r_t^1 \\ b_t^1 \\ s_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ b_t^2 \\ s_t^2 \end{pmatrix}, ...\} \tag{4.14}$$

where landmarks are detected by an onboard active camera in terms of range r_t^i, bearing b_t^i and a signature s_t^i for identifying each landmark. A landmark measurement model is defined by a feature-based map m, which consists of a list of signatures and coordinate locations as follows:

$$m = \{m_1, m_2, ...\} = \{(m_{1,x}, m_{1,y}), (m_{2,x}, m_{2,y}), ...\} \tag{4.15}$$

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

Table 1. Description of variables for obtaining linear, lateral and rotational odometry information.

where the i-th feature at time t corresponds to the j-th landmark detected by a robot whose pose is $x_t = \begin{pmatrix} x & y & \theta \end{pmatrix}^T$ the implemented model is:

$$\begin{pmatrix} r_t^i(x,y,\theta) \\ b_t^i(x,y,\theta) \\ s_t^i(x,y,\theta) \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x}-x)^2 + (m_{j,y}-y)^2} \\ atan2(m_{j,y}-y, m_{j,x}-x)) - \theta \\ s_j \end{pmatrix} \tag{4.16}$$

The proposed landmark model requires an already known environment with defined landmarks and constantly observed visual features. Therefore, robot perception uses mainly defined landmarks if they are qualified as reliable landmarks.

2.2.1 Defined Landmark Recognition

The landmarks are coloured beacons located in a fixed position and are recognised by image operators. Figure 2 presents the quality of the visual detection by a comparison of distance errors in the observations of beacons and goals. As can be seen, the beacons are better recognised than goals when they are far away from the robot. Any visible landmark in a range from 2m to 3m has a comparatively less error than a near object. Figure 2.b shows the angle errors for beacons and goals respectively, where angle errors of beacons are bigger than the ones for goals. The beacon errors slightly reduces when object becomes distant. Contrastingly, the goal errors increases as soon the robot has a wider angle of perception.
These graphs also illustrates errors for observations with distance and angle variations. In both graphs, error measurements are presented in constant light conditions and without occlusion or any external noise that can affect the landmark perception.

2.2.2 Undefined Landmark Recognition

A landmark modelling is used for detecting undefined environment and frequently appearing features. The procedure is accomplished by characterising and evaluating familiarised shapes from detected objects which are characterised by sets of properties or entities. Such process is described in the following stages:

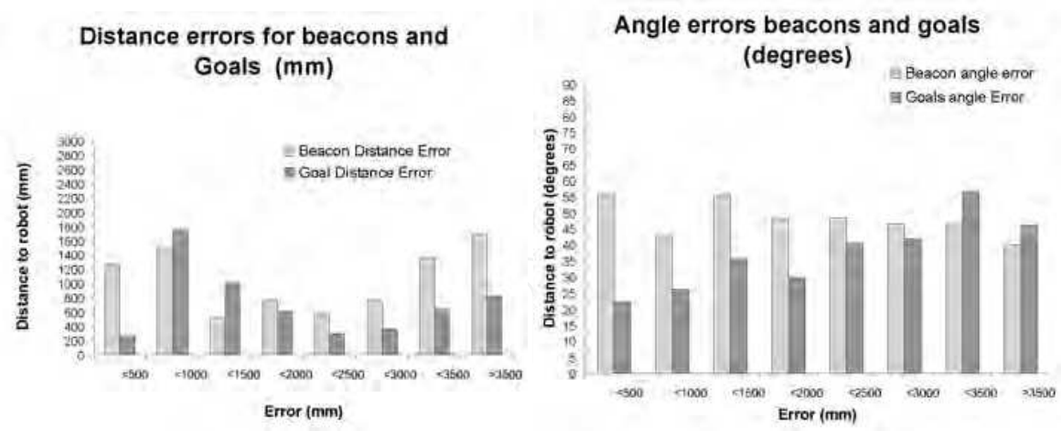


Fig. 2. Distance and angle errors in landmarks observations for beacons and goals of proposed landmark model.

- **Entity Recognition** The first stage of dynamic landmark modelling relies on feature identification from constantly observed occurrences. The information is obtained from colour surface descriptors by a landmark entity structure. An entity is integrated by pairs or triplets of blobs with unique characteristics constructed from merging and comparing linear blobs operators. The procedure interprets surface characteristics for obtaining range frequency by using the following operations:

  1. Obtain and validate entity's position from the robot's perspective.
  2. Get blobs' overlapping values with respect to their size.
  3. Evaluate compactness value from blobs situated in a bounding box.
  4. Validate eccentricity for blobs assimilated in current the entity.

- **Model Evaluation**

  The model evaluation phase describes a procedure for achieving landmark entities for a real time recognition. The process makes use of previously defined models and merges them for each sensing step. The process is described in Algorithm 1:

  From the main loop algorithm is obtained a list of candidate entities $\{E\}$ to obtain a collection of landmark models $\{L\}$. This selection requires three operations for comparing an entity with a landmark model:

  – *Colour combination* is used for checking entities with same type of colours as a landmark model.
  – *Descriptive operators,* are implemented for matching features with a similar characteristics. The matching process merges entities with a ± 0.3 range ratio from defined models.
  – *Time stamp* and *Frequency* are recogised every minute for filtering long lasting models using a removing and merging process of non leading landmark models.

  The merging process is achieved using a bubble sort comparison with a swapping stage modified for evaluating similarity values and it also eliminates 10% of the landmark

Algorithm 1 Process for creating a landmark model from a list of observed features.

Require: Map of observed features $\{E\}$
Require: A collection of landmark models $\{L\}$
{*The following operations generate the landmark model information.*}
for all $\{E\}_i \subseteq \{E\}$ **do**
  Evaluate ColourCombination($\{E\}_i$) $\{C\}_i$
  Evaluate BlobDistances($\{E\}_i$) d_i
  Obtain TimeStamp($\{E\}_i i$) t_i
  Create Entity($\{C\}_i$, d_i, t_i) j
  for $\{L\}_k MATCHON \{L\}$ **do** {*If information is similar to an achieved model* }
    if $j \in \{L\}_k$ **then**
      Update $\{L\}_k(j)$ {*Update modelled values and*}
      Increase $\{L\}_k$ frequency {*Increase modelled frequency*}
    else {*If modelled information does not exist* }
      Create $\{L\}_{k+1}(j)$ {*Create model and*}
      Increase $\{L\}_{k+1}$ frequency {*Increase modelled frequency*}
    end if
    if time > 1 min **then** {*After one minute* }
      MergeList($\{L\}$) {*Select best models*}
    end if
  end for
end for

candidates. The similarity values are evaluated using Equation 3.4 and the probability of perception using Equation 3.5:

$$p(i,j) = \frac{M(i,j)}{\sum_{k=1}^{N} M(k,j)} \tag{3.4}$$

$$M(i,j) = \sum_{l=1}^{P} E(i,j,l) \tag{3.5}$$

where N indicates the achieved candidate models, i is the sampled entity, j is the compared landmark model, $M(i,j)$ is the landmark similarity measure obtained from matching an entity's descriptors and assigning a probability of perception as described in Equation 3.6, P is the total descriptors, l is a landmark descriptor and $E(i,j,l)$ is the Euclidian distance of each landmark model compared, estimated using Equation 3.7:

$$\sum_{k=1}^{N} M(k,j) = 1 \tag{3.6}$$

$$E(i,j,l) = \sqrt{\sum_{m=1}^{L_l} \frac{(i_m - l_m)^2}{\sigma_m^2}} \tag{3.7}$$

where L_l refers to all possible operators from the current landmark model, σ_m is the standard deviation for each sampled entity i_m in a sample set and l is a landmark descriptor value.

3. Localisation Methods

Robot localisation is an environment analysis task determined by an internal state obtained from robot-environment interaction combined with any sensed observations. The traditional state assumption relies on the robot's influence over its world and on the robot's perception of its environment.
Both steps are logically divided into independent processes which use a state transition for integrating data into a predictive and updating state. Therefore, the implemented localisation methods contain measurement and control phases as part of state integration and a robot pose conformed through a Bayesian approach. On the one hand, the control phase is assigned to robot odometry which translates its motion into lateral, linear and rotational velocities. On the other hand, the measurement phase integrates robot sensed information by visual features. The following sections describe particular phase characteristics for each localisation approach.

3.1 Fuzzy Markov Method

As it is shown in the FM grid-based method of (Buschka et al., 2000) and (Herrero-Pérez et al., 2004), a grid G_t contains a number of cells for each grid element $G_t(x, y)$ for holding a probability value for a possible robot position in a range of $[0, 1]$. The fuzzy cell (*fcell*) is represented as a fuzzy trapezoid (Figure 3) defined by a tuple $< \theta, \Delta, \alpha, h, b >$, where θ is robot orientation at the trapezoid centre with values in a range of $[0, 2\pi]$; Δ is uncertainty in a robot orientation θ; h corresponds to fuzzy cell (*fcell*) with a range of $[0, 1]$; α is a slope in the trapezoid, and b is a correcting bias.

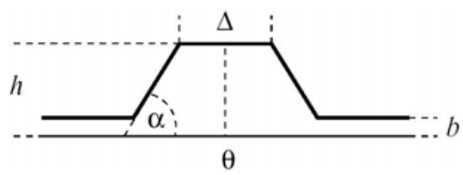


Fig. 3. Graphic representation of robot pose in an *fuzzy cell*

Since a Bayesian filtering technique is implemented, localisation process works in predict-observe-update phases for estimating robot state. In particular, the *Predict* step adjusts to motion information from robot movements. Then, the *Observe* step gathers sensed information. Finally, the *Update* step incorporates results from the *Predict* and *Observe* steps for obtaining a new estimation of a fuzzy grid-map. The process sequence is described as follows:

1. **Predict step**. During this step, robot movements along grid-cells are represented by a distribution which is continuously blurred. As described in previous work in (Herrero-Pérez et al., 2004), the blurring is based on odometry information reducing grid occupancy for robot movements as shown in Figure 4(c)). Thus, the grid state G'_t is obtained by performing a translation and rotation of G_{t-1} state distribution according to motion $\vec{u}$. Subsequently, this odometry-based blurring, B_t, is uniformly calculated for including uncertainty in a motion state.

Thus, state transition probability includes as part of robot control, the blurring from odometry values as it is described in the following equation:

$$G'_t = f(G_t \mid G_{t-1}, \vec{u}) \otimes B_t \tag{4.30}$$

2. **Observe step**. In this step, each observed landmark i is represented as a vector $\vec{z}_i$, which includes both range and bearing information obtained from visual perception. For each observed landmark $\vec{z}_i$, a grid-map $S_{i,t}$ is built such that $S_{i,t}(x, y, \theta|\vec{z}_i)$ is matched to a robot position at (x, y, θ) given an observation $\vec{r}$ at time t.
3. **Update step**. At this step, grid state G'_t obtained from the prediction step is merged with each observation step $S_{t,i}$. Afterwards, a fuzzy intersection is obtained using a product operator as follows:

$$G_t = f(z_t \mid G_t) \tag{4.31}$$

$$G_t = G'_t \times S_{t,1} \times S_{t,2} \times \cdots \times S_{t,n} \tag{4.32}$$

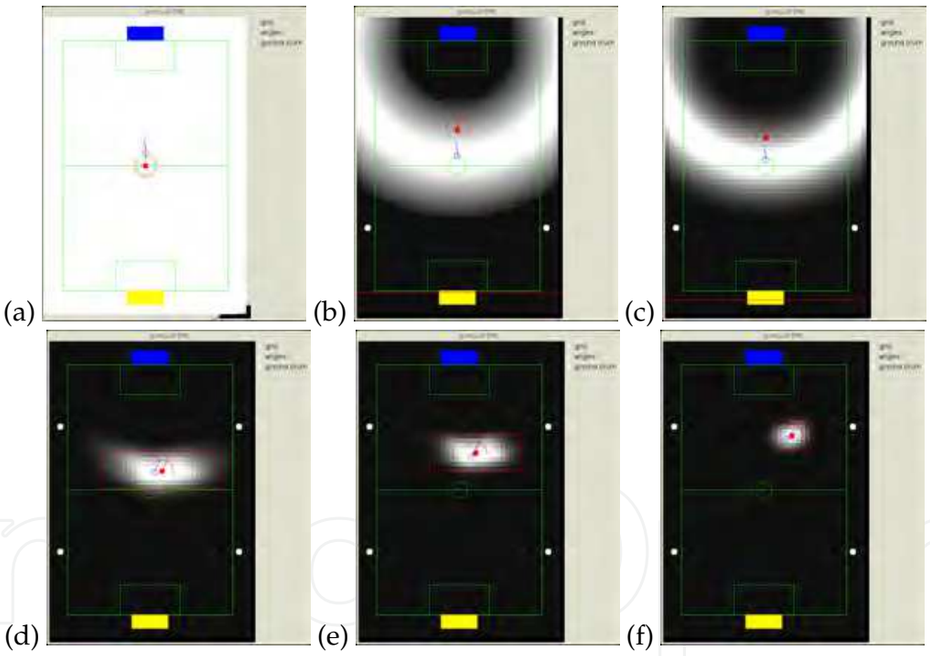


Fig. 4. In this figure is shown a simulated localisation process of FM grid starting from absolute uncertainty of robot pose (a) and some initial uncertainty (b) and (c). Through to an approximated (d) and finally to an acceptable robot pose estimation obtained from simulated environment explained in (Samperio & Hu, 2008).

A simulated example of this process is shown in Figure 4. In this set of Figures, Figure 4(a) illustrates how the system is initialised with absolute uncertainty of robot pose as the white areas. Thereafter, the robot incorporates landmark and goal information where each grid state G_t is updated whenever an observation can be successfully matched to a robot position, as

illustrated in Figure 4(b). Subsequently, movements and observations of various landmarks enable the robot to localise itself, as shown from Figure 4(c) to Figure 4(f).
This method's performance is evaluated in terms of accuracy and computational cost during a real time robot execution. Thus, a reasonable *fcell* size of 20 cm^2 is addressed for less accuracy and computing cost in a pitch space of 500cmx400cm.
This localisation method offers the following advantages, according to (Herrero-Pérez et al., 2004):

- Fast recovery from previous errors in the robot pose estimation and kidnappings.
- Multi-hypotheses for robot pose (x, y) .
- It is much faster than classical Markovian approaches.

However, its disadvantages are:

- Mono-hypothesis for orientation estimation.
- It is very sensitive to sensor errors.
- The presence of false positives makes the method unstable in noisy conditions.
- Computational time can increase dramatically.

3.2 Extended Kalman Filter method

Techniques related to EKF have become one of the most popular tools for state estimation in robotics. This approach makes use of a state vector for robot positioning which is related to environment perception and robot odometry. For instance, robot position is adapted using a vector s_t which contains (x, y) as robot position and θ as orientation.

$$\mathbf{s} = \begin{pmatrix} x_{robot} \\ y_{robot} \\ \theta_{robot} \end{pmatrix} \tag{4.17}$$

As a Bayesian filtering method, EKF is implemented *Predict* and *Update* steps, described in detail below:
1. Prediction step. This phase requires of an initial state or previous states and robot odometry information as control data for predicting a state vector. Therefore, the current robot state s_t^- is affected by odometry measures, including a noise approximation for error and control estimations P_t^-. Initially, robot control probability is represented by using:

$$s_t^- = f(s_{t-1}, u_{t-1}, w_t) \tag{4.18}$$

where the nonlinear function f relates the previous state s_{t-1}, control input u_{t-1} and the process noise w_t.
Afterwards, a covariance matrix P_t^- is used for representing errors in state estimation obtained from the previous step's covariance matrix P_{t-1} and defined process noise. For that reason, the covariance matrix is related to the robot's previous state and the transformed control data, as described in the next equation:

$$P_t^- = A_t P_{t-1} A_t^T + W_t Q_{t-1} W_t^T \tag{4.19}$$

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:

$$A_t = fs = \begin{pmatrix} 1 & 0 & -u_t^{lat} cos\theta_t - u_t^{lin} sen\theta_{t-1} \\ 0 & 1 & u_t^{lin} cos\theta_t - u_t^{lat} sen\theta_{t-1} \\ 0 & 0 & 1 \end{pmatrix} \tag{4.19}$$

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

$$Q_t = E[w_t w_t^T] \tag{4.20}$$

The Sony AIBO robot may not be able to obtain a landmark observation at each localisation step but it is constantly executing a motion movement. Therefore, it is assumed that frequency of odometry calculation is higher than visually sensed measurements. For this reason, control steps are executed independently from measurement states (Kiriy & Buehler, 2002) and covariance matrix actualisation is presented as follows:

$$s_t = s_t^- \tag{4.21}$$

$$P_t = P_t^- \tag{4.22}$$

2. Updating step. During this phase, sensed data and noise covariance P_t are used for obtaining a new state vector. The sensor model is also updated using measured landmarks $m_{1\cdots 6,(x,y)}$ as environmental descriptive data. Thus, each z_t^i of the i landmarks is measured as distance and angle with a vector (r_t^i, ϕ_t^i). In order to obtain an updated state, the next equation is used:

$$s_t = s_{t-1} + K_t^i (z_t^i - \hat{z}_t^i) = s_{t-1} + K_t^i (z_t^i - h^i(s_{t-1})) \tag{4.23}$$

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

$$\hat{z}_t^i = h^i(s_{t-1}) = \begin{pmatrix} \sqrt{(m_{t,x}^i - s_{t-1,x})^2 + (m_{t,y}^i - s_{t-1,y})} \\ atan^2(m_{t,x}^i - s_{t-1,x}, m_{t,y}^i - s_{t-1,y}) - s_{t-1,\theta} \end{pmatrix} \tag{4.24}$$

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

$$K_t^i = P_{t-1} (H_t^i)^T (S_t^i)^{-1} \tag{4.25}$$

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

$$S_t^i = H_t^i P_{t-1} (H_t^i)^T + R_t^i \tag{4.26}$$

Then H_t^i describes changes in the robot position as follows:

$$\begin{aligned} H_t^i &= h^i(s_{t-1}) s_t \\ &= \begin{pmatrix} -\frac{m_{t,x}^i - s_{t-1,x}}{\sqrt{(m_{t,x}^i - s_{t-1,x})^2 + (m_{t,y}^i - s_{t-1,y})^2}} & -\frac{m_{t,y}^i - s_{t-1,y}}{\sqrt{(m_{t,x}^i - s_{t-1,x})^2 + (m_{t,y}^i - s_{t-1,y})^2}} & 0 \\ \frac{m_{t,y}^i - s_{t-1,y}}{(m_{t,x}^i - s_{t-1,x})^2 + (m_{t,y}^i - s_{t-1,y})^2} & -\frac{m_{t,x}^i - s_{t-1,x}}{(m_{t,x}^i - s_{t-1,x})^2 + (m_{t,y}^i - s_{t-1,y})^2} & -1 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned} \tag{4.27}$$

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

$$P_t = (I - K_t^i H_t^i) P_{t-1} \tag{4.28}$$

Finally, as not all $\hat{z}_t^i$ values are obtained at every observation, z_t^i values are evaluated for each observation and δ_t^i is a confidence measurement obtained from Equation (4.29). The confidence observation measurement has a threshold value between 5 and 100, which varies according to localisation quality.

$$\delta_t^i = (z_t^i - \hat{z}_t^i)^T (S_t^i)^{-1} (z_t^i - \hat{z}_t^i) \tag{4.29}$$

3.3 FM-EKF method

Merging the FM and EKF algorithms is proposed in order to achieve computational efficiency, robustness and reliability for a novel robot localisation method. In particular, the FM-EKF method deals with inaccurate perception and odometry data for combining method hypotheses in order to obtain the most reliable position from both approaches.
The hybrid procedure is fully described in Algorithm 2, in which the *fcell* grid size is (50-100 cm) which is considerably wider than FM's. Also the *fcell* is initialised in the space map centre. Subsequently, a variable is iterated for controlling FM results and it is used for comparing robot EKF positioning quality. The localisation quality indicates if EKF needs to be reset in the case where the robot is lost or the EKF position is out of FM range.

Algorithm 2 Description of the FM-EKF algorithm.

Require: $position_{FM}$ over all pitch
Require: $position_{EKF}$ over all pitch
while *robotLocalise* **do**
  $\{Execute"Predict"phases for FM and EKF\}$
  Predict $position_{FM}$ using motion model
  Predict $position_{EKF}$ using motion model
  $\{Execute"Correct"phases for FM and EKF\}$
  Correct $position_{FM}$ using perception model
  Correct $position_{EKF}$ using perception model
  $\{Check quality of localisation for EKF using FM\}$
  if $(quality(position_{FM}) \gg quality(position_{EKF})$ **then**
    Initialise $position_{EKF}$ to $position_{FM}$
  else
    robot position $\leftarrow position_{EKF}$
  end if
end while

The FM-EKF algorithm follows the predict-observe-update scheme as part of a Bayesian approach. The input data for the algorithm requires similar motion and perception data. Thus, the hybrid characteristics maintain a combined hypothesis of robot pose estimation using data that is independently obtained. Conversely, this can adapt identical measurement and control information for generating two different pose estimations where, under controlled circumstances one depends on the other.
From one viewpoint, FM localisation is a robust solution for noisy conditions. However, it is also computationally expensive and cannot operate efficiently in real-time environments

with a high resolution map. Therefore, its computational accuracy is inversely proportional to the *fcell* size. From a different perspective, EKF is an efficient and accurate positioning system which can converge computationally faster than FM. The main drawback of EKF is a misrepresentation in the multimodal positioning information and method initialisation.

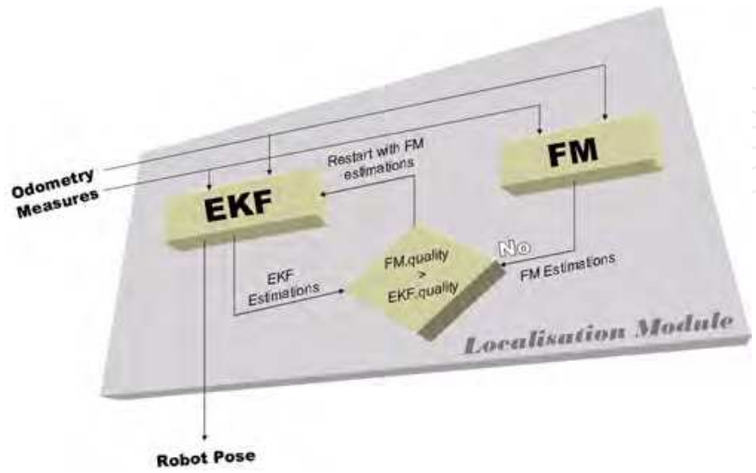


Fig. 5. Flux diagram of hybrid localisation process.

The hybrid method combines FM grid accuracy with EKF tracking efficiency. As it is shown in Figure 5, both methods use the same control and measurement information for obtaining a robot pose and positioning quality indicators. The EKF quality value is originated from the eigenvalues of the error covariance matrix and from noise in the grid- map.
As a result, EKF localisation is compared with FM quality value for obtaining a robot pose estimation. The EKF position is updated whenever the robot position is lost or it needs to be initialised. The FM method runs considerably faster though it is less accurate.
This method implements a Bayesian approach for robot-environment interaction in a localisation algorithm for obtaining robot position and orientation information. In this method a wider *fcell* size is used for the FM grid-map implementation and EKF tracking capabilities are developed to reduce computational time.

4. System Overview

The configuration of the proposed HRI is presented in Figure 6. The user-robot interface manages robot localisation information, user commands from a GUI and the overhead tracking, known as the VICON tracking system for tracking robot pose and position. This overhead tracking system transmits robot heading and position data in real time to a GUI where the information is formatted and presented to the user.
The system also includes a robot localisation as a subsystem composed of visual perception, motion and behaviour planning modules which continuously emits robot positioning information. In this particular case, localisation output is obtained independently of robot behaviour moreover they share same processing resources. Additionally, robot-visual information can be generated online from GUI from characterising environmental landmarks into robot configuration.
Thus, the user can manage and control the experimental execution using online GUI tasks. The GUI tasks are for designing and controlling robot behaviour and localisation methods,

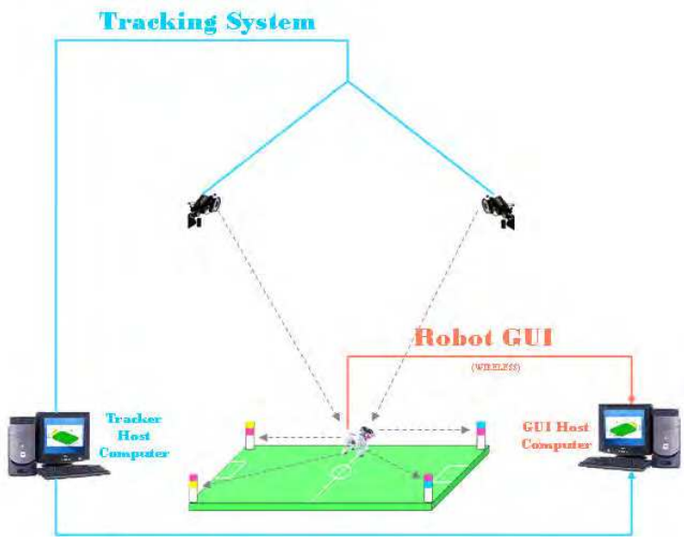


Fig. 6. Complete configuration of used human-robot interface.

and for managing simulated and experimental results. Moreover, tracking results are the experiments' input and output of a grand truth that is evaluating robot self-localisation results.

5. Experimental Results

The presented experimental results contain the analysis of the undefined landmark models and a comparison of implemented localisation methods. The undefined landmark modelling is designed for detecting environment features that could support the quality of the localisation methods. All localisation methods make use of defined landmarks as main source of information.

The first set of experiments describe the feasibility for employing a not defined landmark as a source for localisation. These experiments measure the robot ability to define a new landmark in an indoor but dynamic environment. The second set of experiments compare the quality of localisation for the FM, EKF and FM-EKF independently from a random robot behaviour and environment interaction. Such experiments characterise particular situations when each of the methods exhibits an acceptable performance in the proposed system.

5.1 Dynamic landmark acquisition

The performance for angle and distance is evaluated in three experiments. For the first and second experiments, the robot is placed in a fixed position on the football pitch where it continuously pans its head. Thus, the robot maintains simultaneously a perception process and a dynamic landmark creation. Figure 7 show the positions of 1683 and 1173 dynamic models created for the first and second experiments over a duration of five minutes.

Initially, newly acquired landmarks are located at 500 mm and with an angle of $3\Pi/4 rad$ from the robot's centre. Results are presented in Table **??**. The tables for Experiments 1 and 2, illustrate the mean ($\tilde{x}$) and standard deviation (σ) of each entity's distance, angle and errors from the robot's perspective.

In the third experiment, landmark models are tested during a continuous robot movement. This experiment consists of obtaining results at the time the robot is moving along a circular

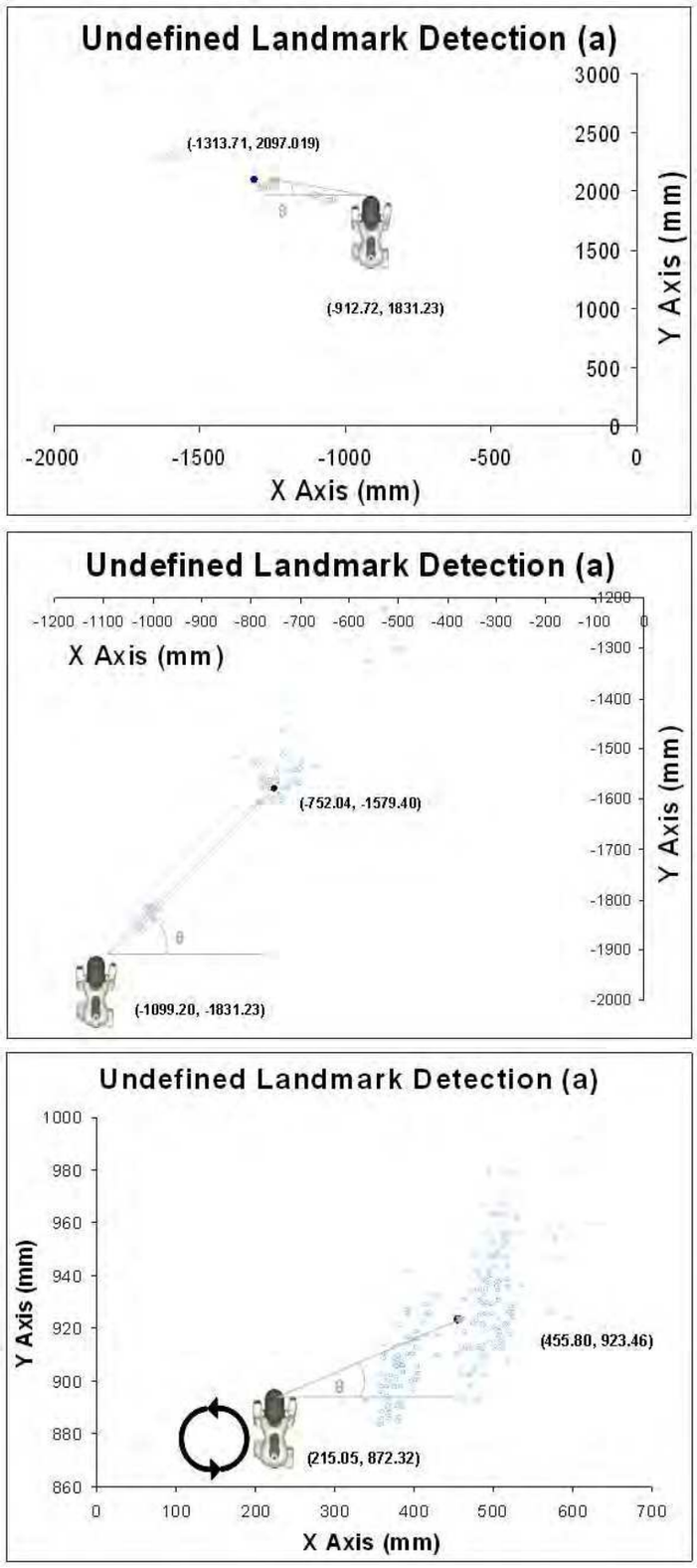


Fig. 7. Landmark model recognition for Experiments 1, 2 and 3

Experpiment 1	**Distance**	**Angle**	**Error in Distance**	**Error in Angle**
Mean	489.02	146.89	256.46	2.37
StdDev	293.14	9.33	133.44	8.91

Experpiment 2	**Distance**	**Angle**	**Error in Distance**	**Error in Angle**
Mean	394.02	48.63	86.91	2.35
StdDev	117.32	2.91	73.58	1.71

Experpiment 3	**Distance**	**Angle**	**Error in Distance**	**Error in Angle**
Mean	305.67	12.67	90.30	3.61
StdDev	105.79	4.53	54.37	2.73

Table 2. Mean and standard deviation for experiment 1, 2 and 3.

trajectory with 20 cm of bandwidth radio, and whilst the robot's head is continuously panning. The robot is initially positioned 500 mm away from a coloured beacon situated at 0 *degrees* from the robot's mass centre. The robot is also located in between three defined and one undefined landmarks. Results obtained from dynamic landmark modelling are illustrated in Figure 7. All images illustrate the generated landmark models during experimental execution. Also it is shown darker marks on all graphs represent an accumulated average of an observed landmark model.

This experiment required 903 successful landmark models detected over five minute duration of continuous robot movement and the results are presented in the last part of the table for Experiment 3. The results show magnitudes for mean ($\tilde{x}$) and standard deviation (σ), distance, angle and errors from the robot perspective.

Each of the images illustrates landmark models generated during experimental execution, represented as the accumulated average of all observed models. In particular for the first two experiments, the robot is able to offer an acceptable angular error estimation in spite of a variable proximity range. The results for angular and distance errors are similar for each experiment. However, landmark modelling performance is susceptible to perception errors and obvious proximity difference from the perceived to the sensed object.

The average entity of all models presents a minimal angular error in a real-time visual process. An evaluation of the experiments is presented in Box and Whisker graphs for error on position, distance and angle in Figure 8.

Therefore, the angle error is the only acceptable value in comparison with distance or positioning performance. Also, the third experiment shows a more comprehensive real-time measuring with a lower amount of defined landmark models and a more controllable error performance.

5.2 Comparison of localisation methods

The experiments were carried out in three stages of work: (i) simple movements; (ii) combined behaviours; and (iii) kidnapped robot. Each experiment set is to show robot positioning abilities in a RoboCup environment. The total set of experiment updates are of 15, with 14123 updates in total. In each experimental set, the robot poses estimated by EKF, FM and FM-EKF localisation 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:

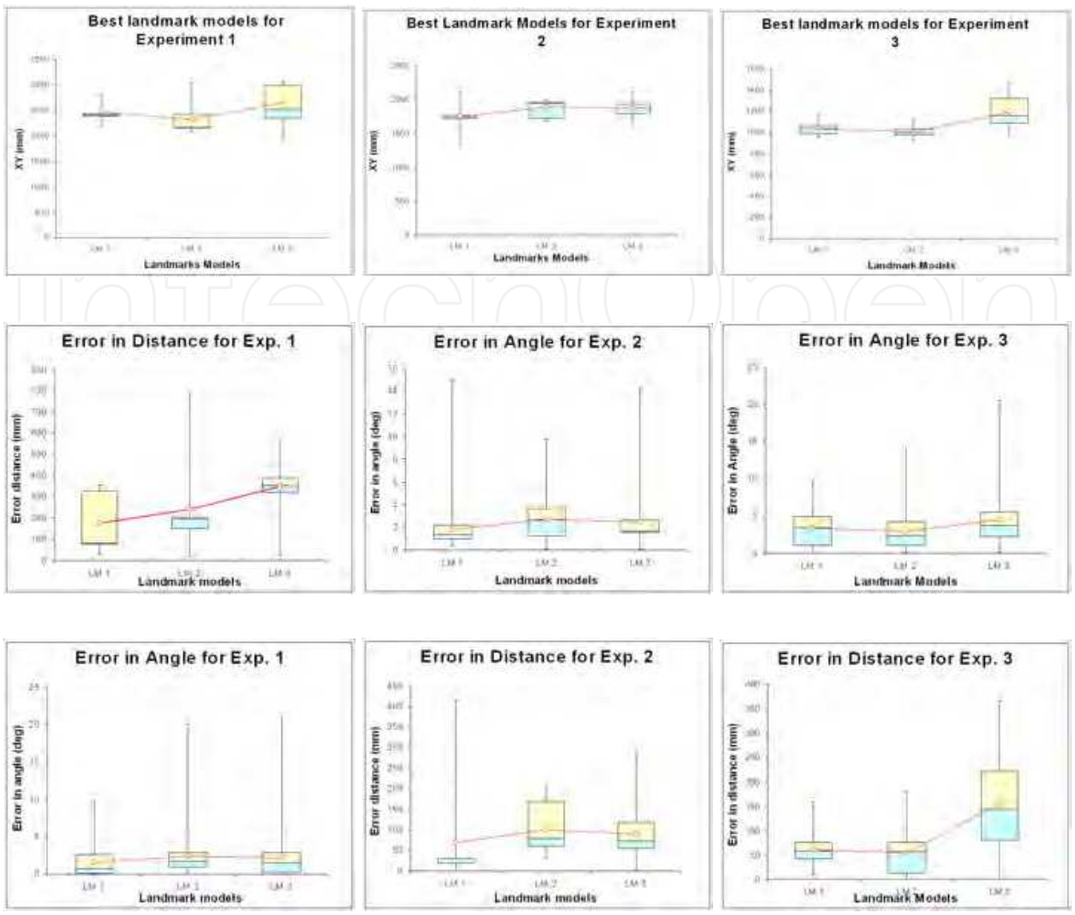


Fig. 8. Error in angle for Experiments 1, 2 and 3.

1. **Simple Movements**. This stage includes straight and circular robot trajectories in forward and backward directions within the pitch.
2. **Combined Behaviour**. This stage is composed by a pair of high level behaviours. Our first experiment consists of reaching a predefined group of coordinated points along the pitch. Then, the second experiment is about playing alone and with another dog to obtain localisation results during a long period.
3. **Kidnapped Robot**. This stage is realised randomly in sequences of kidnap time and pose. For each kidnap session the objective is to obtain information about where the robot is and how fast it can localise again.

All experiments in a playing session with an active localisation are measured by showing the type of environment in which each experiment is conducted and how they directly affect robot behaviour and localisation results. In particular, the robot is affected by robot displacement, experimental time of execution and quantity of localisation cycles. These characteristics are described as follows and it is shown in Table 3:

1. **Robot Displacement** is the accumulated distance measured from each simulated method step from the perspective of the grand truth mobility.

2. **Localisation Cycles** include any completed iteration from update-observe-predict stages for any localisation method.
3. **Time of execution** refers to total amount of time taken for each experiment with a time of 1341.38 s for all the experiments.

	Exp. 1	**Exp. 2**	**Exp. 3**
Displacement (mm)	15142.26	5655.82	11228.42
Time of Execution (s)	210.90	29.14	85.01
Localisation Cycles (iterations)	248	67	103

Table 3. Experimental conditions for a simulated environment.

The experimental output depends on robot behaviour and environment conditions for obtaining parameters of performance. On the one side, robot behaviour is embodied by the specific robot tasks executed such as *localise*, *kick the ball*, *search for the ball*, *search for landmarks*, *search for players*, *move to a point in the pitch*, *start*, *stop*, *finish*, and so on. On the other side, robot control characteristics describe robot performance on the basis of values such as: *robot displacement*, *time of execution*, *localisation cycles* and *landmark visibility*. Specifically, robot performance criteria are described for the following environment conditions:

1. **Robot Displacement** is the distance covered by the robot for a complete experiment, obtained from grand truth movement tracking. The total displacement from all experiments is 146647.75 mm.
2. **Landmark Visibility** is the frequency of the detected true positives for each landmark model among all perceived models. Moreover, the visibility ranges are related per each localisation cycle for all natural and artificial landmarks models. The average landmark visibility obtained from all the experiments is in the order of 26.10 % landmarks per total of localisation cycles.
3. **Time of Execution** is the time required to perform each experiment. The total time of execution for all the experiments is 902.70 s.
4. **Localisation Cycles** is a complete execution of a correct and update steps through the localisation module. The amount of tries for these experiments are 7813 cycles.

The internal robot conditions is shown in Table **??**:

	Exp 1	**Exp 2**	**Exp 3**
Displacement (mm)	5770.72	62055.79	78821.23
Landmark Visibility (true positives/total obs)	0.2265	0.3628	0.2937
Time of Execution (s)	38.67	270.36	593.66
Localisation Cycles (iterations)	371	2565	4877

Table 4. Experimental conditions for a real-time environment.

In Experiment 1, the robot follows a trajectory in order to localise and generate a set of visible ground truth points along the pitch. In Figures 9 and 10 are presented the error in X and Y axis by comparing the EKF, FM, FM-EKF methods with a grand truth. In this graphs it is shown a similar performance between the methods EKF and FM-EKF for the error in X and Y

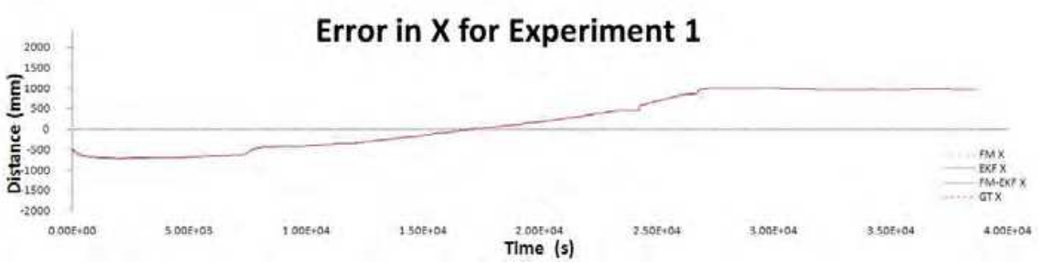


Fig. 9. Error in X axis during a simple walk along the pitch in Experiment 1.

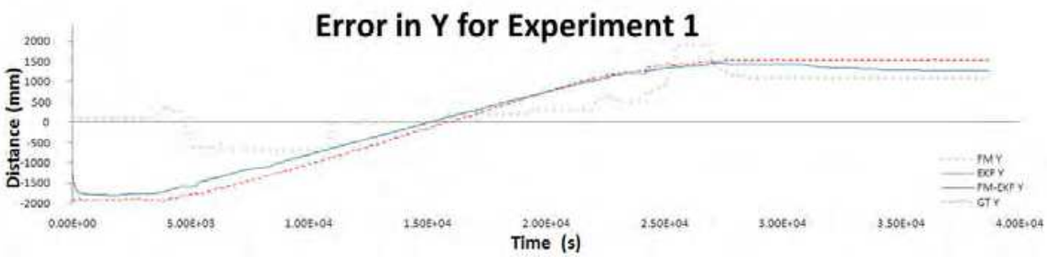


Fig. 10. Error in Y axis during a simple walk along the pitch in Experiment 1.

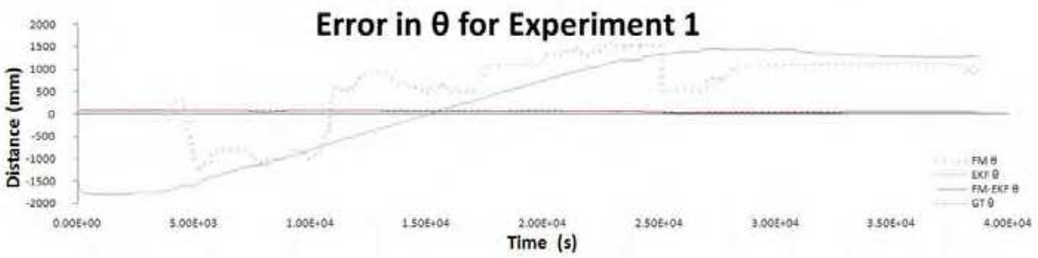


Fig. 11. Error in θ axis during a simple walk along the pitch in Experiment 1.

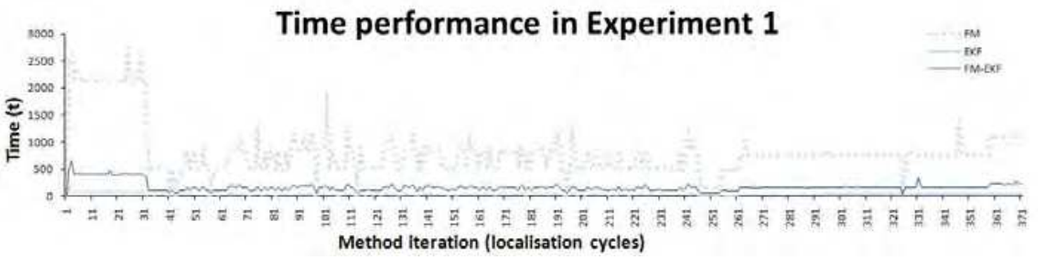


Fig. 12. Time performance for localisation methods during a walk along the pitch in Exp. 1.

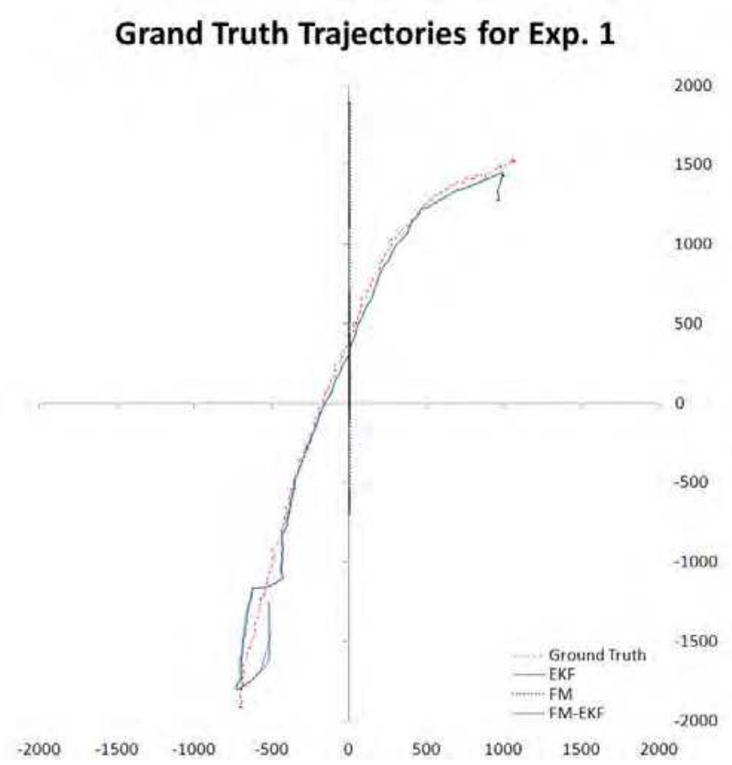


Fig. 13. Robot trajectories for EKF, FM, FM-EKF and the overhead camera in Exp. 1.

axis but a poor performance of the FM. However for the orientation error displayed in Figure 11 is shown that whenever the robot walks along the pitch without any lack of information, FM-EKF improves comparatively from the others. Figure 12 shows the processing time for all methods, in which the proposed FM-EKF method is faster than the FM method, but slower than the EKF method. Finally, in Figure 13 is presented the estimated trajectories and the overhead trajectory. As can be seen, during this experiment is not possible to converge accurately for FM but it is for EKF and FM-EKF methods where the last one presents a more accurate robot heading.

For Experiment 2, is tested a combined behaviour performance by evaluating a playing session for a single and multiple robots. Figures 14 and 15 present as the best methods the EKF and FM-EKF with a slightly improvement of errors in the FM-EKF calculations. In Figure 16 is shown the heading error during a playing session where the robot visibility is affected by a constantly use of the head but still FM-EKF, maintains an more likely performance compared to the grand truth. Figure 17 shows the processing time per algorithm iteration during the robot performance with a faster EKF method. Finally, Figure 18 shows the difference of robot trajectories estimated by FM-EKF and overhead tracking system.

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 localisation performance is evaluated and shown in the figures for Experiment 3. Figures 19 and 20 show positioning errors for X and Y axis during a kidnapping sequence. Also, FM-EKF has a similar development for orientation error as it is shown in Figure 21. Figure 22 shows the processing

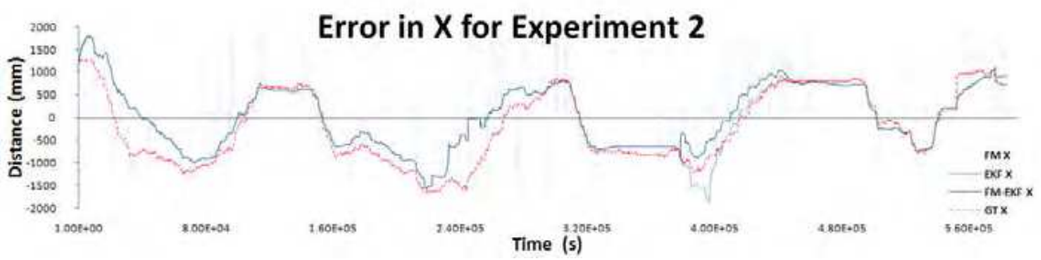


Fig. 14. Error in X axis during a simple walk along the pitch in Experiment 2.

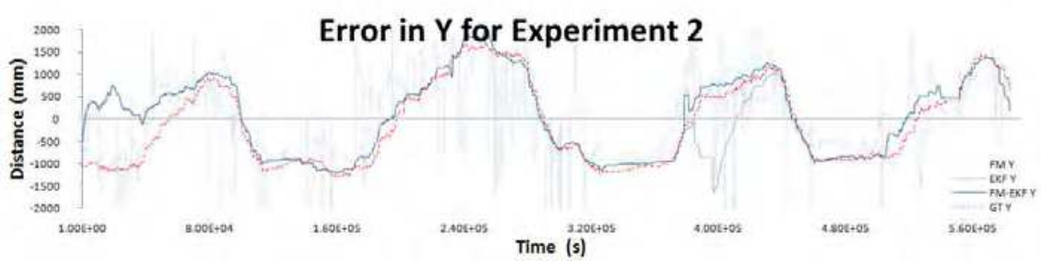


Fig. 15. Error in Y axis during a simple walk along the pitch in Experiment 2.

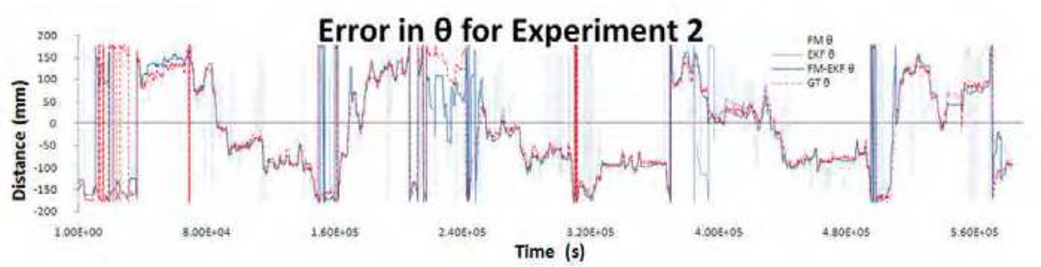


Fig. 16. Error in θ axis during a simple walk along the pitch in Experiment 2.

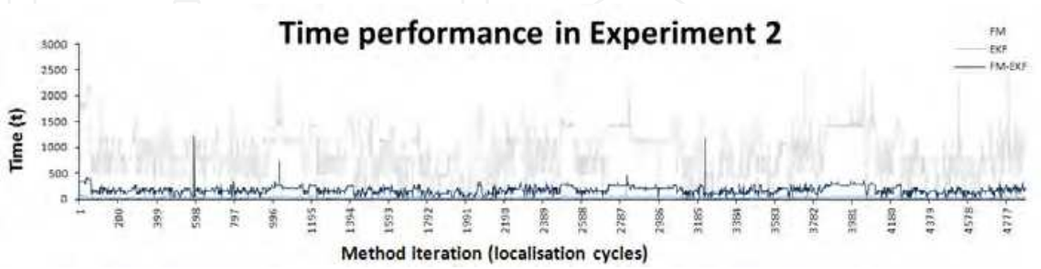


Fig. 17. Time performance for localisation methods during a walk along the pitch in Exp. 2.

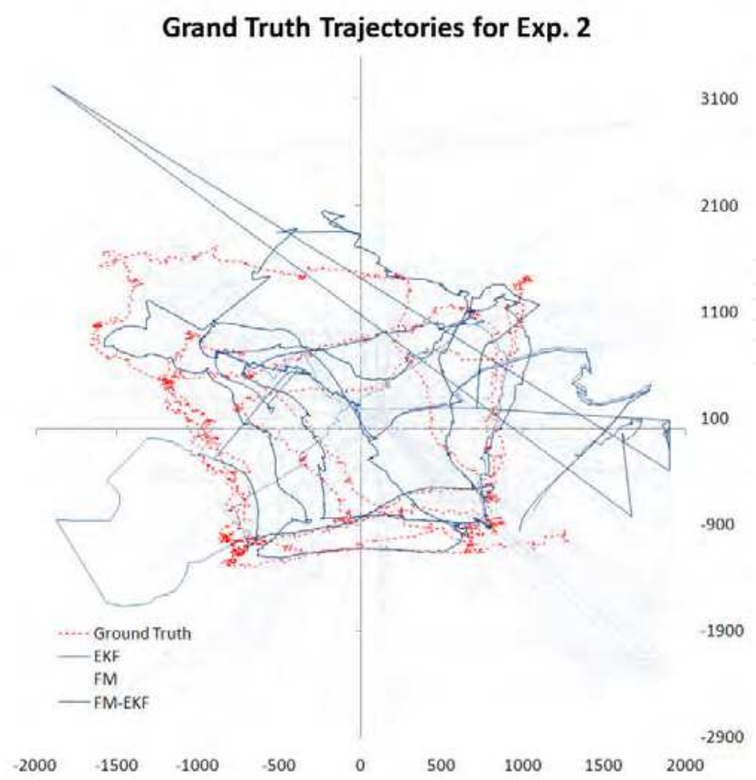


Fig. 18. Robot trajectories for EKF, FM, FM-EKF and the overhead camera in Exp. 2.

time per iteration for all algorithms a kidnap session. Finally, in Figure 23 and for clarity reasons is presented the trajectories estimated only by FM-EKF, EKF and overhead vision system. Results from kidnapped experiments show the resetting transition from a local minimum to fast convergence in 3.23 seconds. Even EKF has the most efficient computation time, FM-EKF offers the most stable performance and is a most suitable method for robot localisation in a dynamic indoor environment.

6. Conclusions

This chapter presents an implementation of real-time visual landmark perception for a quadruped walking robot in the RoboCup domain. The proposed approach interprets an object by using symbolic representation of environmental features such as natural, artificial or undefined. Also, a novel hybrid localisation approach is proposed for fast and accurate robot localisation of an active vision platform. The proposed FM-EKF method integrates FM and EKF algorithms using both visual and odometry information.

The experimental results show that undefined landmarks can be recognised accurately during static and moving robot recognition sessions. On the other hand, it is clear that the hybrid method offers a more stable performance and better localisation accuracy for a legged robot which has noisy odometry information. The distance error is reduced to ± 20 mm and the orientation error is 0.2 degrees.

Further work will focus on adapting for invariant scale description during real time image processing and of optimising the filtering of recognized models. Also, research will focus on

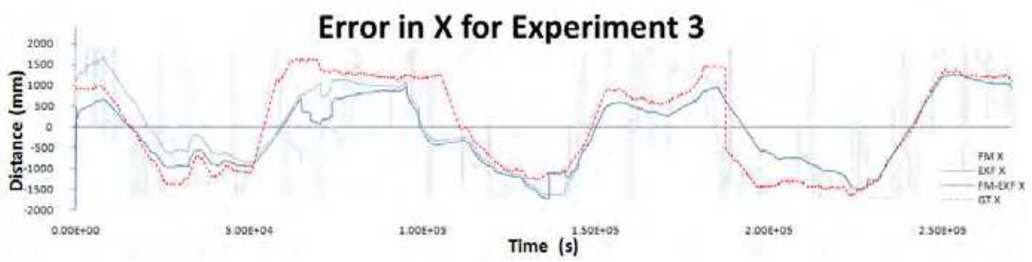


Fig. 19. Error in X axis during a simple walk along the pitch in Experiment 3.

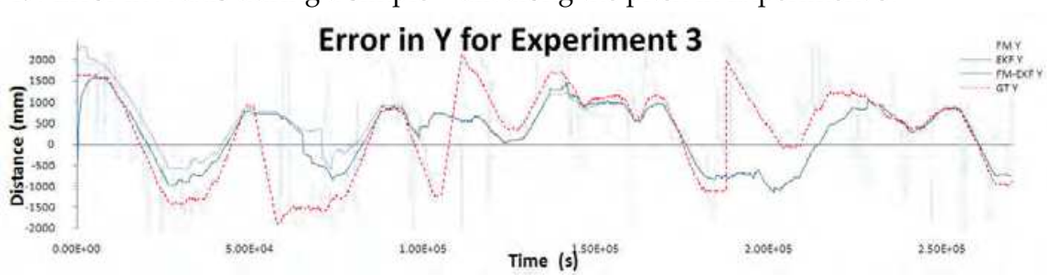


Fig. 20. Error in Y axis during a simple walk along the pitch in Experiment 3.

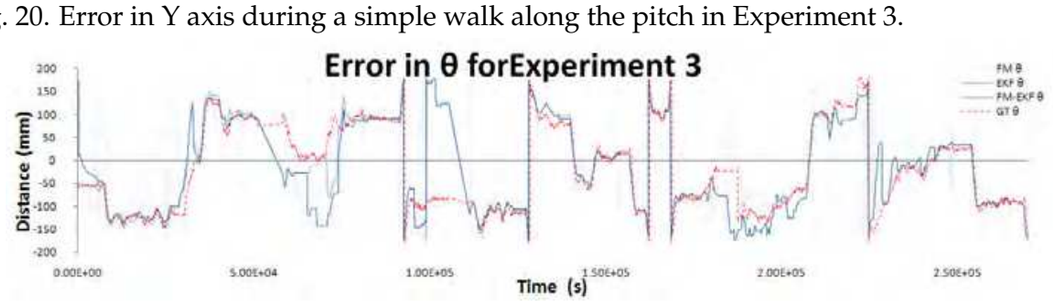


Fig. 21. Error in θ axis during a simple walk along the pitch in Experiment 3.

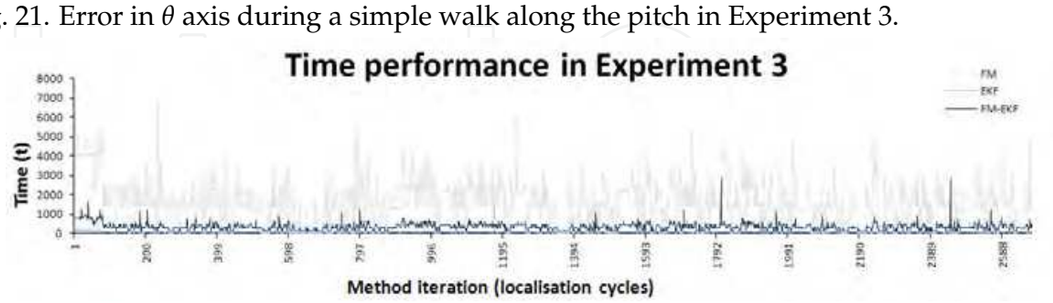


Fig. 22. Time performance for localisation methods during a walk along the pitch in Exp. 3.

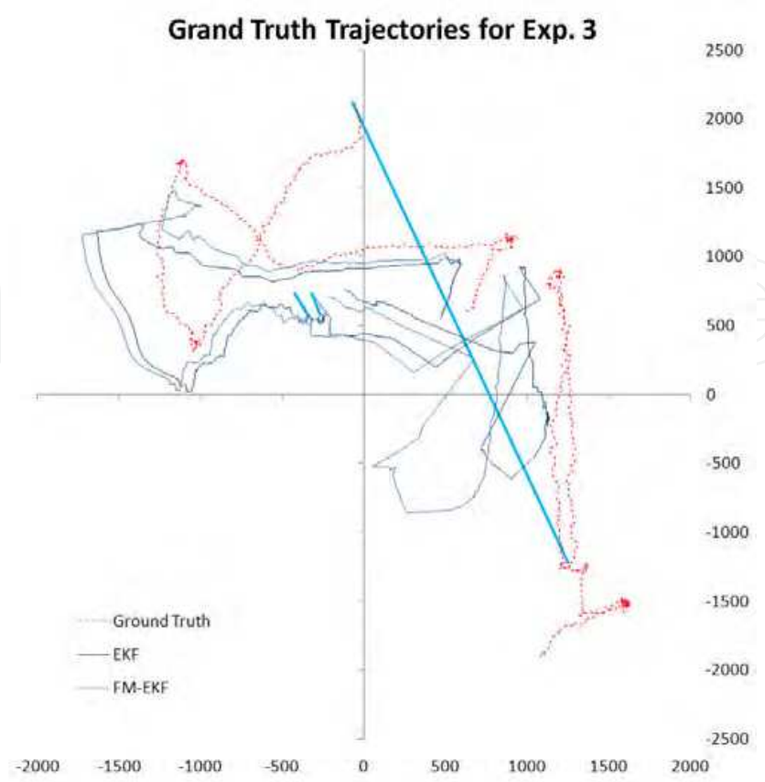


Fig. 23. Robot trajectories for EKF, FM, FM-EKF and the overhead camera in Exp. 3., where the thick line indicates kidnapped period.

the reinforcement of the quality in observer mechanisms for odometry and visual perception, as well as the improvement of landmark recognition performance.

7. Acknowledgements

We would like to thank TeamChaos (http://www.teamchaos.es) for their facilities and programming work and Essex technical team (http://essexrobotics.essex.ac.uk) for their support in this research. Part of this research was supported by a CONACyT (Mexican government) scholarship with reference number 178622.

8. References

Baltzakis, H. & Trahanias, P. (2002). Hybrid mobile robot localization using switching state-space models., *Proc. of IEEE International Conference on Robotics and Automation*, Washington, DC, USA, pp. 366–373.

Buschka, P., Saffiotti, A. & Wasik, Z. (2000). Fuzzy landmark-based localization for a legged robot, *Proc. of IEEE Intelligent Robots and Systems*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1205 – 1210.

Duckett, T. & Nehmzow, U. (2000). Performance comparison of landmark recognition systems for navigating mobile robots, *Proc. 17th National Conf. on Artificial Intelligence (AAAI-2000)*, AAAI Press/The MIT Press, pp. 826 – 831.

Fox, D., Burgard, W., Dellaert, F. & Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots, *AAAI/IAAI*, pp. 343–349.
URL: *http://citeseer.ist.psu.edu/36645.html*

Fox, D., Burgard, W. & Thrun, S. (1998). Markov localization for reliable robot navigation and people detection, *Sensor Based Intelligent Robots*, pp. 1–20.

Gutmann, J.-S. (2002). Markov-kalman localization for mobile robots., *ICPR (2)*, pp. 601–604.

Gutmann, J.-S., Burgard, W., Fox, D. & Konolige, K. (1998). An experimental comparison of localization methods, *Proc. of IEEE/RSJ International Conference on Intelligen Robots and Systems*, Vol. 2, pp. 736 – 743.

Hatice, K., C., B. & A., L. (2006). Comparison of localization methodsfor a robot soccer team, *International Journal of Advanced Robotic Systems* **3**(4): 295–302.

Hayet, J., Lerasle, F. & Devy, M. (2002). A visual landmark framework for indoor mobile robot navigation, *IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3942 – 3947.

Herrero-Pérez, D., Martínez-Barberá, H. M. & Saffiotti, A. (2004). Fuzzy self-localization using natural features in the four-legged league, *in* D. Nardi, M. Riedmiller & C. Sammut (eds), *RoboCup 2004: Robot Soccer World Cup VIII*, LNAI, Springer-Verlag, Berlin, DE. Online at http://www.aass.oru.se/˜asaffio/.

Jensfelt, P., Austin, D., Wijk, O. & Andersson, M. (2000). Feature based condensation for mobile robot localization, *Proc. of IEEE International Conference on Robotics and Automation*, pp. 2531 – 2537.

Kiriy, E. & Buehler, M. (2002). Three-state extended kalman filter for mobile robot localization, *Technical Report TR-CIM 05.07*, McGill University, Montreal, Canada.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I. & Osawa, E. (1997). Robocup: The robot world cup initiative, *International Conference on Autonomous Agents archive*, Marina del Rey, California, United States, pp. 340 – 347.

Kristensen, S. & Jensfelt, P. (2003). An experimental comparison of localisation methods, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 992 – 997.

Luke, R., Keller, J., Skubic, M. & Senger, S. (2005). Acquiring and maintaining abstract landmark chunks for cognitive robot navigation, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2566– 2571.

Maosen, W., Hashem, T. & Zell, A. (2005). Robot navigation using biosonar for natural landmark tracking, *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 3 – 7.

Quoc, V. D., Lozo, P., Jain, L., Webb, G. I. & Yu, X. (2004). A fast visual search and recognition mechanism for real-time robotics applications, *Lecture notes in computer science* **3339**(17): 937–942. XXII, 1272 p.

Samperio, R. & Hu, H. (2008). An interactive HRI for walking robots in robocup, *In Proc. of the International Symposium on Robotics and Automation IEEE*, ZhangJiaJie, Hunan, China.

Samperio, R. & Hu, H. (2010). Implementation of a localisation-oriented HRI for walking robots in the robocup environment, *International Journal of Modelling, Identification and Control (IJMIC)* **12**(2).

Samperio, R., Hu, H., Martin, F. & Mantellan, V. (2007). A hybrid approach to fast and accurate localisation for legged robots, *Robotica* . Cambridge Journals (In Press).

Sung, J. A., Rauh, W. & Recknagel, M. (1999). Circular coded landmark for optical 3d-measurement and robot vision, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2, pp. 1128 – 1133.

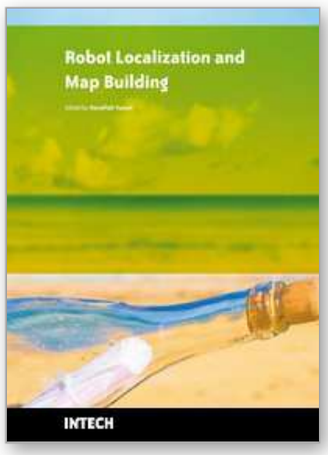
Tanaka, K., Kimuro, Y., Okada, N. & Kondo, E. (2004). Global localization with detection of changes in non-stationary environments, *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1487 – 1492.

Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J. & Schulz, D. (2000). Probabilistic algorithms and the interactive museum tour-guide robot minerva, *International Journal of Robotics Research* **19**(11): 972–999.

Thrun, S., Fox, D., Burgard, W. & Dellaert, F. (2001). Robust monte carlo localization for mobile robots, *Journal of Artificial Intelligence* **128**(1-2): 99–141.
URL: *http://citeseer.ist.psu.edu/thrun01robust.html*

Watman, C., Austin, D., Barnes, N., Overett, G. & Thompson, S. (2004). Fast sum of absolute differences visual landmark, *IEEE International Conference on Robotics and Automation*, Vol. 5, pp. 4827 – 4832.

Yoon, K. J. & Kweon, I. S. (2001). Artificial landmark tracking based on the color histogram, *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, Vol. 4, pp. 1918–1923.



Robot Localization and Map Building

Edited by Hanafiah Yussof

ISBN 978-953-7619-83-1
Hard cover, 578 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

Localization and mapping are the essence of successful navigation in mobile platform technology. Localization is a fundamental task in order to achieve high levels of autonomy in robot navigation and robustness in vehicle positioning. Robot localization and mapping is commonly related to cartography, combining science, technique and computation to build a trajectory map that reality can be modelled in ways that communicate spatial information effectively. This book describes comprehensive introduction, theories and applications related to localization, positioning and map building in mobile robot and autonomous vehicle platforms. It is organized in twenty seven chapters. Each chapter is rich with different degrees of details and approaches, supported by unique and actual resources that make it possible for readers to explore and learn the up to date knowledge in robot navigation technology. Understanding the theory and principles described in this book requires a multidisciplinary background of robotics, nonlinear system, sensor network, network engineering, computer science, physics, etc.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen