July 23-27 2007, Vienna, Austria
www.indin2007.org

INDIN 2007

CONFERENCE PROCEEDINGS VOLUME 1

5th IEEE International Conference on Industrial Informatics

SIEMENS

TU VIENNA Technische Universität Wien Vienna University of Technology

OVE

oneida

IEEE

ICS

Comparison of Self-Localization Methods for Soccer Robots

Abdul Bais,*Member, IEEE,*Tobias Deutsch, *Member, IEEE,*Gregor Novak, *Member, IEEE*

***Abstract*— This paper presents a comparison of two localization algorithms for tiny autonomous robots in a well known but highly dynamic environment. Position knowledge is gained through collecting information about distinctive features in the environment with a stereo vision system and comparing it to a model of the world. The model consists of a map of the static environment and information about moving objects. Based upon this model, the sensor data is used to generate a hypothesis of the position of the robot in the real world. A robot soccer game played by small, autonomous robots is the test-bed for this work. Constraints such as small size of the robot and the dynamic nature of the environment has to be taken into account while developing any solution for the localization. Simulation results show that for the given application, the extended Kalman filter based method is comparable in performance to the particle filter based method, although the particle filter has high time complexity.**

I. Introduction

With the passage of time, robots are becoming more and more important for private homes [1]. Therefore, they need to evolve from industry oriented applications to machines that are useful for households. Part of this evolution will be the development of the ability to act autonomously in a chaotic environment. The industrial environments are normally fixed. On contrary the environment in private homes in often chaotic and the robots have to adapt to it. Autonomy not only include the ability to make strategic decision but also self-positioning without any external help.

Although localization could be a simple task of collecting all data, merging it and figuring out the perfect position, however, one is confronted with problems which can be divided into two groups. The first group consists of "fundamental problems", which are due to restrictions in hardware or stem from inherent computer science (complexity) limitations. Sensors are not reliable and they cannot have a complete view of the environment. Computational power is limited and algorithms have to be selected carefully. Furthermore, if two sensor readings are made at different local times, they need to be integrated sequentially. Alternative approaches to this problem can be found in [2].

The second group can be roughly termed as "environmental conditions", which are located in the real world. These problems are design issues and are a priori decided if the robot has to deal with them. An important problem of this group is the kidnapped robot problem. Here the robot is arbitrarily displaced and it has to assess the situation and thereafter performs a global relocation [3]. A special case of the kidnapped robot problem is the bootstrap problem where the robot is told that it is being displaced [3]. Furthermore, there could be unknown moving objects around the robot or the static environment may change. For a changing environment the robot has to deal with the map building problem and localization simultaneously.

A. Bais, T. Deutsch, and G. Novak are with the Institute of Computer Technology of the Vienna University of Technology: phone +43 1 58801 38421, fax +43 1 58801 38499; (e-mail: {surname}@ict.tuwien.ac.at)

Majority of localization algorithms start at a known position and add internal movement data (e.g. odometric sensor data) and external environment data (e.g. distance sensors) to this position [4], [6], [7]. These algorithms are usually not able to recover from tracking failures in cyclic environments. Global position estimation is required at startup and when/if the robot fails to track its position.

The Kalman Filter (KF) and its non-linear versions have been extensively applied to robot navigation problems [8], [9]. Tsai [10] reports a Digital Signal Processor (DSP) based multi-sensor integration for localization using Extended Kalman Filter (EKF). The dead-reckoning sub-system merges data from magnetic compass, a rate-gyroscope and two magnetic encoders. The ultrasonic system provide range measurements and consists of one ultrasonic and one radio frequency controlled switch mounted at a known location fixed to an inertial frame of reference, while four ultrasonic receivers and radio frequency controlled switch installed on the mobile robot.

Madhavan and Durrant-Whyte [11] report a system in which points of maxima are extracted from laser scan data. These point landmarks are combined in an EKF to estimate localization of an outdoor vehicle operating in relatively unmodified environments. Gutmann et al. [12] generate a set position hypotheses from laser range matching. They select the most plausible one and fuse it with odometry position estimate using an EKF.

Due to its unimodal nature, KF is not suitable for global localization in environments where features are not distinctive in the global space. To solve this problem some researchers generate and track multiple hypothesis to constitute a framework for global KF [13]. Other approaches to global self-localization include Markov Localization (ML) [14] and Monte Carlo Localization (MCL) methods [15]. These methods are capable of recovering a robot from tracking failures and can deal with multi-modal densities. ML model represent probability densities at a fine resolution and perform numerical integration over a grid of points [8]. This approach is very general but, due to its generality, inefficient. Even with optimization techniques, the time needed to process the grid is prohibitive. Another problem is the a priori commitment to the size of the state space.

A more efficient approach, MCL, represent probability density by a set of samples that are randomly drawn from it. Fichtner and Grossmann [16] reports a sensor model and its use in a MCL frame work for camera based pose estimation. Similarly, samples of the particle filter are weighted using the properties of the Fourier transform of omnidirectional images according to the similarity among images [17].

The robot environment is assumed to be well known but highly dynamic and cannot be modified. Two different types of features are found in the environment: they are line based and color transitions [18]. Line based features are extracted using gradient based Hough transform, whereas, color transitions between different colored patches are detected using segmentation methods.

Due to the scarcity of landmarks our global position estimation algorithm is based on minimum number of distinct landmarks. Two distinct landmarks are required to estimate a global position if it is possible to order these landmarks with respect to the robot [19]. This requirement, however, falls by one if the absolute orientation of the robot is available [20]. However, instantaneous acquisition of distinct landmarks at all times is not possible as landmarks are few and are frequently occluded. Therefore, it is required to track the robot position.

The simulated environment is designed to be similar to the robot soccer MiroSOT league [21]. The robot in the simulation is build after the robot Tinyphoon [22], which is equipped with a stereo vision system and onboard processors to allow autonomous action.

In this paper we focus on position estimation comparison using an EKF and MCL. To study the convergence of robot position belief when features are available we initialize the robot with a random starting position that is selected with a uniform probability over the robot state space. The balance of the paper is structured as follows: Section II formulates the localization algorithm in the framework of an EKF, discusses the robot control vector and robot observation vector. Uncertainty of the control vector and robot observation is also presented. Section III illustrates state estimation using MCL or particle filters. Experimental results are presented in Section IV and finally the paper is concluded in Section V.

II. Extended Kalman Filter

The KF is a recursive state estimator, which can deal with incomplete and noisy data [23], [24]. In its original form KF can only be applied to linear systems with Gaussian error models. In order to apply to non-linear systems, they have to be linearized first. EKF is one of the non-linear KF [9].

The model for EKF is described by (1) and (2). The first one defines the recursive state estimator $\mathbf{p}_k \in \mathbb{R}^n$, where n is the dimension of the state vector and k is the time index. The non-linear function $\mathbf{f}$ models the transition from a previous state $\mathbf{p}_{k-1}$ to the next state $\mathbf{p}_k$ under the influence of the system input $\mathbf{u}_k \in \mathbb{R}^o$. The superscript o is a constant which gives the dimension of $\mathbf{u}_k$. The control vector is independent of state $\mathbf{p}_{k-1}$ and is supposed to be corrupted by an additive zero mean Gaussian noise $\widetilde{\mathbf{u}}_k$ of covariance $\mathbf{U}_k$.

$$\mathbf{p}_k = \mathbf{f}(\mathbf{p}_{k-1}, \mathbf{u}_k, k) \quad (1)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{p}_k, k) + \mathbf{w}_k \quad (2)$$

Equation (2) links the state vector $\mathbf{p}_k$ with the sensor observation vector $\mathbf{z}_k \in \mathbb{R}^m$, where m is the dimension of the sensor vector. This relation is modeled by the non-linear function $\mathbf{h}$. Imperfection of the observation model is represented by $\mathbf{w}_k$, which is assumed to be zero mean Gaussian of strength $\mathbf{R}$.

The algorithm for EKF is listed in Algorithm 1. It is divided into two phases which are processed periodically. In the first phase called the "time update" or the "prediction phase", system state at k is predicted based upon the previous state estimation $\mathbf{p}_{k-1|k-1}$ and the estimated value of the system input or control vector $\widehat{\mathbf{u}}_k$. The prediction (3) and its uncertainty (5) is derived using multivariate Taylor series expansion of $\mathbf{p}_k$ around $\widehat{\mathbf{p}}_{k-1|k-1}$ and $\widehat{\mathbf{u}}_k$, and is a minimum mean square estimate of $\mathbf{p}_k$ conditioned on all sensor measurements but the one at time k. In these derivations it is assumed that the state estimation error $\widetilde{\mathbf{p}}_{k-1|k-1}$ is zero mean and independent of $\widetilde{\mathbf{u}}_k$.

Algorithm 1 Extended Kalman Filter Algorithm [25]

upon initialization **do**
  input $\widehat{\mathbf{p}}_{0|0}$ and $\mathbf{P}_{0|0}$
  $k \leftarrow 0$
end upon
loop
  $k \leftarrow k+1$
  if new control input available **then**
    // time update stage
    // predicting future position and its uncertainty

$$\widehat{\mathbf{p}}_{k|k-1} \leftarrow \mathbf{f}(\widehat{\mathbf{p}}_{k-1|k-1}, \widehat{\mathbf{u}}_k, k) \quad (3)$$

$$\mathbf{U}_k \leftarrow \mathbf{J}_u \mathbf{\Sigma}_v \mathbf{J}_u^T \quad (4)$$

$$\mathbf{P}_{k|k-1} \leftarrow \mathbf{J}_1 \mathbf{P}_{k-1|k-1} \mathbf{J}_1^T + \mathbf{J}_2 \mathbf{U}_k \mathbf{J}_2^T \quad (5)$$

$$\mathbf{z}_{k|k-1} \leftarrow \mathbf{h}(\widehat{\mathbf{p}}_{k|k-1}, k) \quad (6)$$

  else

$$\widehat{\mathbf{p}}_{k|k-1} \leftarrow \widehat{\mathbf{p}}_{k-1|k-1}$$
$$\mathbf{P}_{k|k-1} \leftarrow \mathbf{P}_{k|k-1}$$

  end if
  if new observation available **then**
    // updating position based on new observation
    // measurement update stage

$$\mathbf{K} \leftarrow \mathbf{P}_{k|k-1} \mathbf{J}_{zp} \mathbf{S}^{-1} \quad (7)$$

$$\mathbf{R} \leftarrow \mathbf{J}_z \mathbf{\Sigma}_i \mathbf{J}_z^T \quad (8)$$

$$\widehat{\mathbf{p}}_{k|k} \leftarrow \widehat{\mathbf{p}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{z}_{k|k-1}) \quad (9)$$

$$\mathbf{P}_{k|k} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{J}_{zp})\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}\mathbf{J}_{zp})^T + \mathbf{K}\mathbf{R}\mathbf{K}^T \quad (10)$$

  else

$$\widehat{\mathbf{p}}_{k|k} \leftarrow \widehat{\mathbf{p}}_{k|k-1}$$
$$\mathbf{P}_{k|k} \leftarrow \mathbf{P}_{k|k-1}$$

  end if
end loop
where

$$\mathbf{J}_1 = \frac{\partial \mathbf{f}}{\partial \mathbf{p}}, \mathbf{J}_2 = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}, \mathbf{J}_u = \frac{\partial \mathbf{u}}{\partial \mathbf{d}}, \mathbf{\Sigma}_d = \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} \quad (11)$$

$$\mathbf{S} = \mathbf{J}_{zp} \mathbf{P}_{k|k-1} \mathbf{J}_{zp}^T + \mathbf{R}, \mathbf{J}_{zp} = \frac{\partial \mathbf{h}}{\partial \mathbf{p}}, \mathbf{\Sigma}_i = \sigma_{uu}^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (12)$$

As given by (5), the uncertainty of the prediction is increased by the process noise $\mathbf{J}_2 \mathbf{U}_k \mathbf{J}_2^T$. Here $\mathbf{J}_1$ and $\mathbf{J}_2$ are the jacobian of (1) w.r.t $\mathbf{p}_{k-1}$ and $\mathbf{u}_k$ evaluated at $\widehat{\mathbf{p}}_{k-1|k-1}$ and $\widehat{\mathbf{u}}_k$ respectively. The control vector $\mathbf{u}_k$ is defined as action taken that cause a state change in robot frame of reference and given by the following transformation [26]:

$$\mathbf{u}_k = \begin{bmatrix} \delta x_k \\ \delta y_k \\ \delta \theta_k \end{bmatrix} = \begin{bmatrix} \frac{b(d_{rk}+d_{lk})}{2(d_{rk}-d_{lk})} sin(\frac{d_{rk}-d_{lk}}{b}) \\ \frac{b(d_{rk}+d_{lk})}{2(d_{rk}-d_{lk})}(1 - cos(\frac{d_{rk}-d_{lk}}{b})) \\ \frac{d_{rk}-d_{lk}}{d} \end{bmatrix} \quad (13)$$

where d_{lk} and d_{rk} denote the distance covered by the left and right wheels of the robot and b is the wheel base. It is assumed that the deviation $\widetilde{\mathbf{d}}_k$ of the estimated distance vector $\widehat{\mathbf{d}}_k$ from its true value $\mathbf{d}_k$ is a random vector of zero mean and covariance $\mathbf{\Sigma}_d$. The variances of d_{lk} and d_{rk} are denoted by σ_l^2 and σ_r^2, respectively, and are proportional to their absolute values. This error is propagated to $\mathbf{u}_k$ by (13). Using first order approximations of (13) the expression (4) for $\mathbf{U}_k$ is derived. $\mathbf{J}_u$ is the jacobian of $\mathbf{u}_k$ with respect to $\mathbf{d}_k$ evaluated at $\widehat{\mathbf{d}}_k$.

The second phase, also known as the "measurement update" or the "correction phase", corrects prediction using the noisy measurements. The Kalman gain $\mathbf{K}$ (7) gives the proportion between how much the predicted state can be trusted, and up to which extent the new measurements can be taken into account in a way that results in a minimum mean square estimation error [25]. In (7) $\mathbf{J}_{zp}$ is the jacobian of (2) w.r.t $\mathbf{p}_k$ evaluated at $\widehat{\mathbf{p}}_{k|k-1}$ and $\mathbf{S}$ is the innovation covariance matrix. Innovation represent the deviation of the actual observation from the predicted, a measure of disagreement between the actual and predicted state, and is used to correct the predicted state estimate in conjunction with the Kalman gain as given by (9). The covariance of this estimate is given by (10). $\mathbf{I}$ is the identity matrix with dimension $n \times n$. The observation uncertainty $\mathbf{R}$ is based on the assumption that errors in estimating landmark location in the left and right camera are independent identically distributed Gaussian random variables having zero mean and covariance matrix $\mathbf{\Sigma}_i$, where σ_{uu}^2 is the variance of estimating single point in an image.

III. Monte Carlo Localization

As stated earlier, the MCL can only be optimized to some extent. With the growth of the environment either the precision of the state space representation or the calculation time becomes problematic. The solution is to reduce the number of cells representing the state space without deteriorating the required precision. The concept of MCL—or particle filter—is to replace the dense grid representation by a set of particles. A particle is a state vector representing the belief that the robot is at certain location.

The MCL algorithm is based on a Monte Carlo approximation with Sequential Importance Sampling (SIS). The algorithm is initialized with randomly distributed particles. A particle stores the belief in its weight. This is initialized with the reciprocal value of the number of particles. During each cycle, the current weight of a particle is determined by multiplying the probability that the particle represents the real position of the robot in accordance with the current sensor readings. Next, the weights of the particles are normalized. Finally, the new state is estimated using the particles and their weights.

The SIS step — multiplying the old weight with the new probability — leads to degeneration. Particles with a low probability are converging towards zero after several cycles. This results in the problem that strong particles will dominate the overall belief even if it does not conform with the sensor readings. To avoid this, SIS could be replaced with Sampling Importance Resampling (SIR). With SIR, the new weight is not based on the previous weight of the particle. After the state estimation, a new set of particles is drawn from the current set. This new set is used as basis for the next cycle. Particles with a high probability are more likely drawn. This results in areas with high probability gaining weight by replication of particles. Less probable areas are thinning out. This introduces another source of degeneration due to lack of appropriate particles as local maxima dominate the estimation. Further details can be found in [27].

To avoid this type of degeneration, some particles have to be replaced by newly generated uniformly distributed ones. Thereafter, empty areas are refilled regularly. This process is called reinjection. There are several approaches to decide when and how many particles should be replaced. A common approach is to reinject a fixed number of particles each round. Another possibility is to reinject when/if the quality of the estimation is below a threshold.

Reduction in number of particles can be achieved by dynamically adopting the number of particles [28] or using techniques such as Rao-Blackwellization [29]. All of these optimizations are always a tradeoff between required computation power and the number of particles. The number for a given application should be selected carefully.

Algorithm 2 shows a classical particle filter with SIR as the only enhancement. During initialization, all of the N particles are generated by distributing them uniformly over the state space as given by (14). At the beginning of every cycle, the cycle index k is incremented and temporary particles $\tilde{\mathbf{p}}_k{}^{(i)}$ are generated out as given by (15). Using the actual sensor data $\mathbf{z}_k$, the weight $\tilde{w}_k{}^{(i)}$ for the particles is calculated in (16). In (18), the normalized weights of (17) are used to calculate the state estimation for $\mathbf{p}_k$ conditioned on all previous measurements $\mathbf{z}_{1:k}$. As a preparation for the re-sampling step, all tuples $\left\{\tilde{\mathbf{p}}_k{}^{(i)}, \tilde{w}_k{}^{(i)}\right\}$ are sorted in descending order with the highest weight value first. During re-sampling, N times a random number j is drawn in such a way that tuples with a higher weight are preferred. For each j, the corresponding $\tilde{\mathbf{p}}_k{}^{(j)}$ is drawn and assigned to the final state $\mathbf{p}_k^{(i)}$ as described by (19). All final state vectors are assigned a uniform weight. To avoid local maxima, finally, T particles are replaced with new uniformly distributed random particles ($T \ll N$).

IV. Experimental results

The algorithms were compared in a simulated environment. The robot size and dimensions of the playground are set according to the rules outlined by Fira MiroSOT [30]. The only external sensor is the stereo vision system. The robots are also equipped with the odometry. The robot is traveling along a predefined path, hence, the comparison is only focused on passive localization approaches. The comparison criteria is to test if the algorithm is able to address the bootstrap problem and find the global position of the robot. For this experiment we define that the global position is found if the distance to the true position stays less than 12.5 cm for at least three consecutive steps. Thus, the estimated position is inside an area of approximately $1/40^{th}$ of the total possible area.

Fig. 1 sketches the basic components of the environment. It consists of the borders surrounding the field and the four

Algorithm 2 Particle Filter Algorithm

upon initialization **do**
  // generate random particles
  for all $i \in N$ **do**

$$\mathbf{p}_0^{(i)} \sim p(\mathbf{p}_0) \tag{14}$$

  end for
  $k \leftarrow 0$
end upon
upon sensor update y **do**
  $k \leftarrow k + 1$
  for all $i \in N$ **do**
    // generate particle

$$\tilde{\mathbf{p}_k}^{(i)} \leftarrow p\left(\mathbf{p}_k | \mathbf{u}_k, \mathbf{p}_{k-1}^{(i)}\right) \tag{15}$$

    // calculate weight/probability of particle

$$\tilde{w_k}^{(i)} \leftarrow p\left(\mathbf{z}_k | \tilde{\mathbf{p}_k}^{(i)}\right) \tag{16}$$

  end for
  // normalize particle weights
  for all $i \in N$ **do**

$$\tilde{w_k}^{(i)} \leftarrow \tilde{w_k}^{(i)} \left[\sum_{j=1}^{N} \tilde{w_k}^{(i)}\right]^{-1} \tag{17}$$

  end for
  // estimate current state

$$E(\mathbf{p}_k | \mathbf{z}_{1:k}) \leftarrow \sum_{j=1}^{N} \tilde{\mathbf{p}}_k^{(j)} \tilde{w}_k^{(j)} \tag{18}$$

  // re-sample particles
  sort $\left\{\tilde{\mathbf{p}_k}^{(i)}, \tilde{w_k}^{(i)}\right\}_{i=1}^{N}$ such that $\tilde{w_k}^{(i)} > \tilde{w_k}^{(i+1)}$
  for all $i \in N$ **do**
    draw j with probability $\tilde{w_k}$

$$\left\{\mathbf{p}_k^{(i)}, \frac{1}{N}\right\} \leftarrow \left\{\tilde{\mathbf{p}_k}^{(j)}, \tilde{w_k}^{(j)}\right\} \tag{19}$$

  end for
  // reinject random particles
  for all $i \in T$ **do**
    replace $\mathbf{p}_k^{(i)}$ with new random particle
  end for
end upon

goal corners which are used as landmarks. The simulated robot travels a precise circle with a radius of 50cm. The center of the playground is also the center of the path. The robot starts at the right hand side and collects sensor data on its round trip. This trip is divided into 100 steps. After each step, new odometry data is available. In 28 of the 100 steps, at least one landmark is visible. Two landmarks were visible in four of these steps. The first landmark sighting is in step 32. Due to the fact that the robot travels on a precise circular path and performs each step the same relative movement, a landmark is visible in the same steps in every run of the experiment. The observations consists of range and bearing with respect to landmarks [26]. The landmarks are detected using a rule based color segmentation algorithm [18], [19]. The simulated odometric sensor adds noise to the measured distance traveled by each wheel of the robot. The resultant error in the control vector is calculated by compounding the noise vector with the current position.

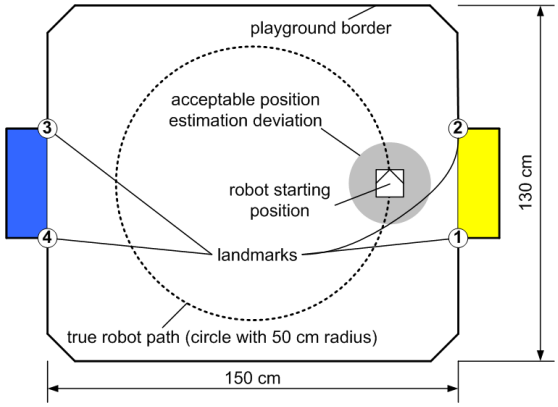

Fig. 1. Simulated Environment

Five different sets of sensor readings are used for this experiment. The algorithms are using random values and hence a single run is not precisely repeated. The results presented here are generated by using 25 different runs for the same input data set. A typical path estimation is shown in Fig. 2. The distance from the estimated position to the true position for each step is shown in Fig. 3.

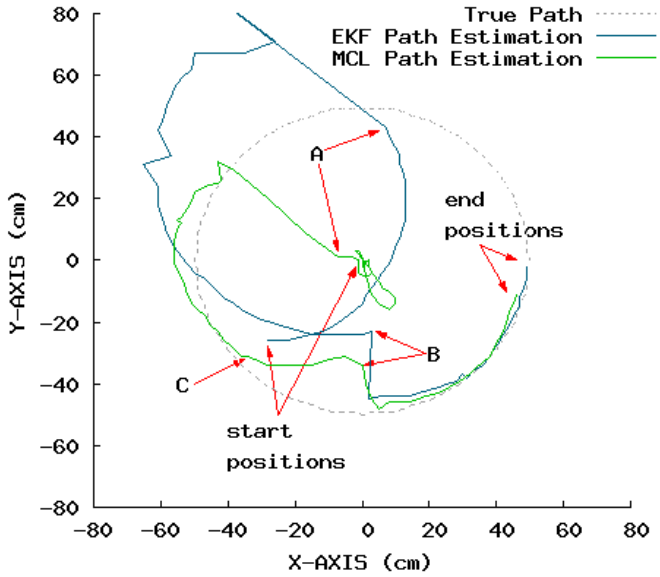

Fig. 2. Path Estimation for one Dataset

The path followed and error for the EKF algorithm is shown by the blue line in Fig. 2 and Fig. 3, respectively. The EKF algorithm is initialized with an arbitrary starting position with high uncertainty. The only restriction is that it is inside the playground. During the first thirty two steps only odometric data is available, which result in a relative movement leading into the wrong direction. After the first sighting of landmark at position marked as 'A', the estimated position jumps to the upper left of the playground. The consecutive landmark sightings do not improve the estimation as the odometric data is dominant until the first sighting of the goal on the other side is made. This position is marked as 'B' in the figure. The path estimation done by the EKF is corrected by the new sightings

towards the real path, which fits the true path well until the end of the round.

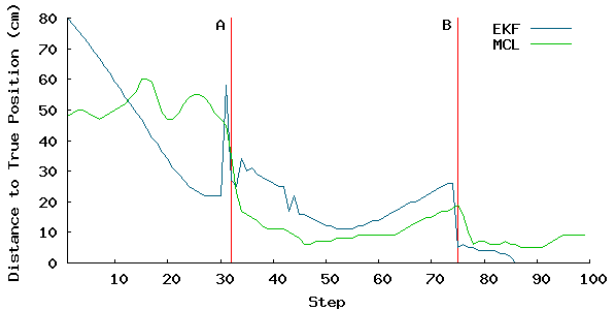

Fig. 3. Distance between Path Estimation and True Path

The robot path as estimated by MCL algorithm and its error is shown by the green line in Fig. 2 and Fig. 3, respectively. The algorithm is initialized with a set of 500 uniformly distributed position estimations (particles). Therefore, the starting point is close to the origin of the coordinate system, which is also the center of the playground. The particle movement at each step is according to the odometric data. Along with the particle resampling/reinjection, the estimated position stays close to the center. With the first landmark sighting at position marked with 'A', the estimation gets closer to the true path. By the time the robot reached the point marked with 'C', fifteen steps without a landmark sighting were passed. Thus, the particle-cloud covers a large area and overlaps the borders of the playground. Particles outside the playground are given a lower probability then the ones inside. Thereafter, instead of continuing the circle, the estimated path is moving parallel to the lower border. Similar to the EKF, with the first landmark sighting of the other goal at point marked as 'B', the estimated path approaches the true path. The final position is about 10 cm behind the true position.

The estimation done by the EKF algorithm is smoother as compared with the one achieved with the MCL. As shown in Fig. 2, the path of the EKF is following a similar path as the true one. An individual particle also follows a path based on the control input, however, the resultant path estimate of the robot does not look like a circular arc as N circular paths contribute to it.

Fig. 3 shows that both algorithms are performing well. Five steps after the first landmark sighting, both estimations are within a range of 30 cm. This improves until the 50^{th}step. Starting with step 80 until the end of the cycle, the estimations are inside a range of 10 cm.

To collect more evidence we repeated the described simulation run for five different sets of data. Fig. 4 shows the performance of the EKF algorithm. The black line denotes the average distance to the true position for all runs in the experiment. The red and the blue lines are marking the area covering 95% of the estimations. Finally, the yellow and the light blue line are showing the upper and the lower bound of the values. This graphs shown in this figure verify the assumption that the estimation stays close to the true path after the first sighting of the second goal.

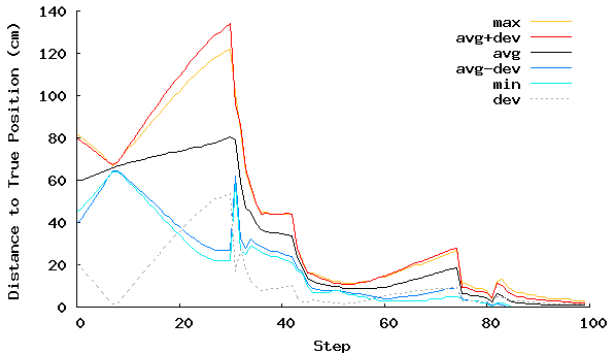

Fig. 4. EKF Experiment Results

Similarly, Fig. 5 shows the results for all five data sets using MCL. Here the same color scheme is used as in Fig. 4. With each set 25 simulation runs have been performed. The average path estimation is close to the one achieved with the EKF, however, the maximum distance to the true position is larger. This happens due to the fact that for a low number of particles it is likely that a certain landmark sighting vector does not fit to any particle. Further experiments with 3000 particles have produced a maximum distance of 39 cm and a standard deviation of 6.5 cm for the last 60 steps. Using 500 particles, these values are 49 cm for the maximum distance and 11.2 cm for the standard deviation.

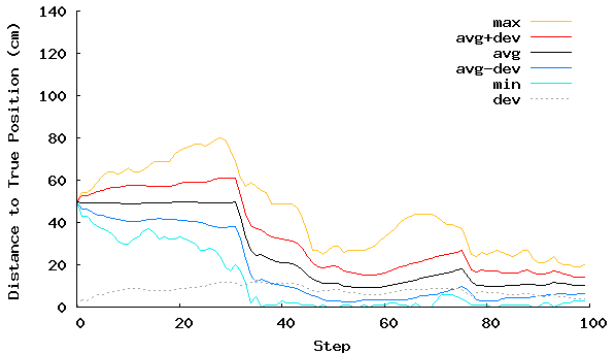

Fig. 5. MCL Experiment Results

Comparison results of the EKF algorithm and the MCL algorithm for the last 20 steps of the cycle are shown in Table I. Both the algorithms are getting closer to the true values. The closeness of the average values during this segment of the cycle is remarkable. The two categories where the EKF is performing better than the MCL are the maximum distance and the standard deviation. Furthermore, the table also shows these numbers for the steps 40—100. Due to the lack of observations between steps 50 and 70 and the larger distance to the landmarks are the reasons for high values in this table.

V. Conclusion

For this application, both—EKF and MCL—are performing well. The slightly better results of the EKF together with the better path shape and the faster performance [31] makes it the localization algorithm of choice for this application (robot soccer).

TABLE I

STATISTICS FOR PARTS OF THE PATH

	steps	min	max	avg.	std.dev	est. criteria met
EKF	40—100	0.0	33.0	12.1	9.3	65%
MCL	40—100	0.0	49.0	12.4	11.2	63%
EKF	80—100	2.0	17.0	9.5	4.9	100%
MCL	80—100	0.0	27.0	10.5	7.3	100%

REFERENCES

[1] P. Tesler, "Universal robots: the history and workings of robotics," May 2005, [Online http://www.thetech.org/exhibits/online/robotics/universal/; accessed 11-January-2007].

[2] C. Kwok, D. Fox, and M. Meil, "Real-time particle filters," *PROCEEDINGS OF THE IEEE*, vol. 92, no. 3, pp. 469–484, MARCH 2004.

[3] S. Engelson and D. McDermott, "Error correction in mobile robot map learning," *Robotics and Automation*, vol. 3, pp. 2555–2560, 1992.

[4] A. Motomura, T. Matsuoka, and T. Hasegawa, "Self-localization method using two landmarks and dead reckoning for autonomous mobile soccer robots," in *RoboCup 2003: Robot Soccer World Cup VII*, ser. LNCS, 2003, pp. 526–533.

[5] G. Adorni, S. Cagnoni, and M. Mordonini, "Landmark-based robot self-localization: a case study for the robocup goal-keeper," in *Proceedings of the International Conference on Information Intelligence and Systems*, Bethesda, MD , USA, October 1999, pp. 164–171.

[6] F. de Jong, J. Caarls, R. Bartelds, and P. Jonker, "A two-tiered approach to self-localization," in *RoboCup 2001: Robot Soccer World Cup V*. Springer-Verlag, 2002, pp. 405–410.

[7] A. F. Tehrani, R. Rojas, H. R. Moballegh, I. Hosseini, and P. Amini, "Analysis by synthesis, a novel method in mobile robot self-localization," in *RoboCup 2004: Robot Soccer World Cup VIII*, ser. LNCS, G. K. et al., Ed., vol. 3276, 2005, pp. 586–593.

[8] D. Fox and S. Thrun, *Probabilistic Robotics*. MIT Press, 2006, iSBN 0-262-20162-3.

[9] R. Negenborn, "Robot localization and kalman filters. on finding your position in a noisy world," Master's thesis, UTRECHT UNIVERSITY, Institute of Information and Computing Sciences, 2003.

[10] C. Tsai, "A localization system of a mobile robot by fusing dead-reckoning and ultrasonic measurements," *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, vol. 47, no. 5, pp. 1399–1406, OCTOBER 1998.

[11] R. Madhavan and H. Durrant-Whyte, "Natural landmark-based autonomous vehicle navigation," *Robotics and Autonomous Systems*, vol. 46, pp. 79–95, 2004.

[12] J. S. Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, T. Weigel, and B. Welsch, "The CS Freiburg robotic soccer team: Reliable self-localization, multirobot sensor integration, and basic soccer skills," in *RoboCup-98: Robot Soccer World Cup II*, M. Asada, Ed. Berlin, Heidelberg, New York: Springer-Verlag, 1999.

[13] K. O. Arras, J. A. Castellanos, M. Schilt, and R. Siegwart, "Feature-based multi-hypothesis localization and tracking using geometric constraints," *Robotics and Autonomous Systems*, vol. 44, pp. 41 – 53, 2003.

[14] D. Fox, "Markov localization: A probabilistic framework for mobile robot localization and navigation," Ph.D. dissertation, Institute of Computer Science III, University of Bonn, Germany, December 1998.

[15] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots(icra' 99)," in *IEEE International Conference on Robotics and Automation*, 1999.

[16] M. Fichtner and A. Grossmann, "A visual-sensor model for mobile robot localisation," Artificial Intelligence Institute, Department of Computer Science, Technische Universitaet Dresden, Dresden, Germany, Tech. Rep. WV-03-03/CL-2003-02, 2003.

[17] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Image-based monte carlo localisation with omnidirectional images," *Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 17–30, 2004.

[18] A. Bais, R. Sablatnig, and G. Novak, "Line-based landmark recognition for self-localization of soccer robots," in *IEEE International Conference on Emerging Technologies (ICET '05)*, Islamabad, Pakistan, September 2005, pp. 132–137.

[19] A. Bais and R. Sablatnig, "Landmark based global self-localization of mobile soccer robots," in *Computer Vision ACCV 2006: 7th Asian Conference on Computer Vision*, ser. LNCS, P. J. Narayanan, S. K. Nayar, and H.-Y. Shum, Eds., vol. 3852. Hyderabad, India: Springer-Verlag GmbH, January 2006, pp. 842 – 851.

[20] A. Bais, R. Sablatnig, J. Gu, and S. Mahlknecht, "Active single landmark based global localization of autonomous mobile robots," in *Proceedings of 2nd International Conference on Visual Computing (ISVC 2006)*, ser. LNCS, G. B. et al., Ed., no. 4291. Lake Tahoe, Nevada, USA: Springer-Verlag Berlin Heidelberg, November 2006, pp. 202 – 211.

[21] FIRA MiroSot, "Overview: FIRA MiroSot small league," http://www.fira.net/soccer/mirosot/overview.html, 2007, [Online; accessed 21-February-2007].

[22] G. Novak and S. Mahlknecht, "Tinyphoon - a tiny autonomous mobile robot," *Proceedings of the IEEE International Symposium on Industrial Electronics 2005*, 2004.

[23] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep., 1995.

[24] P. S. Maybeck, *Stochastic models, estimation, and control*, ser. Mathematics in Science and Engineering. Academic Press, 1979, vol. 141, iSBN 0124807038.

[25] P. Newman, *An Introduction to Estimation and its Application to Mobile Robotics*, 2nd ed., Robotics Research Group, Department of Engineering Science, University of Oxford, October 2005, lecture notes.

[26] A. Bais, R. Sablatnig, J. Gu, and Y. M. Khawaja, "Location tracker for a mobile robot," sumitted to 5th International IEEE Conference on Industrial Informatics (INDIN'07).

[27] T. Deutsch, "Geometric world model repository and localization for autonomous mobile robots," Master's thesis, Technische Universität Wien, Institut für Computertechnik, 2007, to be published.

[28] D. Fox, "Adapting the Sample Size in Particle Filters Through KLD-Sampling," *The International Journal of Robotics Research*, vol. 22, no. 12, pp. 985–1003, 2003.

[29] A. Giremus, A. Doucet, V. Calmettes, and J.-Y. Tourneret, "A rao-blackwellized particle filter for ins/gps integration," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 3, May 2004, pp. 964–967.

[30] FIRA, "Fira small league mirosot game rules," 2006, Online http://www.fira.net/soccer/mirosot/rules_slm.html; accessed 2-December-2006.

[31] J. Gutmann and D. Fox, "An experimental comparison of localization methods continued," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

Abdul Bais was born in 1976 in Shangla, Pakistan. After receiving his Masters in Electrical Engineering from N.W.F.P University of Engineering and Technology, Peshawar Pakistan, he started his PhD at Institute of Computer Technology, Vienna University of Technology, Vienna, Austria, in 2004. During his PhD he spent one year at Robotics Research Laboratory of Dalhousie University Canada, where he focused on information fusion for robot navigation. His research interests include statistical pattern recognition, robot vision and embedded systems.

Tobias Deutsch (M'06) was born 1975 in Vienna/Austria. After graduating at the HTL Ungargasse he started studying Computer Science at the TU Vienna in 1996 and completed his degree in March 2007. Since October 2005 he is staff member of the ICT. His current research focuses on cognitive intelligence, building automation and autonomous agents on the Vienna University of Technology at the Institute of Computer Technology. He is IEEE Member, Organizing Committee member of the INDIN 2007 and Treasurer of the IEEE Student Branch, Section Austria.

Gregor Novak was born 1973 in Vienna. He received a Master's degree in mechanical engineering and a doctoral degree in technical sciences both from the Vienna University of Technology in Austria in the years 1997 and 2002, respectively as well as a Master's degree in engineering management in 2001 from Oakland University in USA. Presently he is the coordinator of the Vienna University of Technology's Center of Excellence for Autonomous Systems (CEAS). His research focuses on autonomous mobile cooperating robots.