A Hybrid Localization Method for a Soccer Playing Robot

Meisam Teimouri[1], Mostafa E. Salehi[1], and Mohammad Reza Meybodi[2]
[1]Mechatronics Research Lab, Islamic Azad University, Qazvin Branch, Qazvin, Iran
[2] Computer Engineering Dep., Amir Kabir University of Technology, Tehran, Iran
{m.teimouri, m.e.salehi}@qiau.ac.ir

***Abstract*—self-localization is the process of estimating the robot position exploiting noisy measurements. Since localization is a key issue for soccer playing robots, some probabilistic approaches have been developed over last years to address it. Methods based on Monte Carlo Localization (MCL) show good ability in dealing with kidnap problem, however, most of them are unstable with limited number of samples. On the other hand, Kalman filter extensions are among the best light weight estimators for position tracking. Their drawback is that they are unimodal and can't be used for global and kidnaped problems. Combining the advantages of these two approaches can lead to a valuable method. In this paper we propose a new hybrid localization method that utilizes the MCL and UKF to reach a stable, multimodal, and low weight localization method. The advantages of our method are evaluated in several experiments.**

***Keywords*—Self-localization; Soccer Playing Robot; Monte Carlo Localization; Unscented Kalman Filter; Kidnap;**

I. Introduction

Self-Localization is the problem of estimating robot position in a known environment. This problem is a key component in autonomous soccer playing robots. A Robot who is not aware of its position cannot make proper decisions. To achieve this goal, a robot should access to relative and absolute measurements [1]. Relative measurements contain information about the location with respect to the previous state. In soccer playing robots, these measurements are usually gathered by odometry. Absolute measurements are acquired from the environment with sensors like camera and contain information about the robot pose independent of prior estimation. Given these measurements, the robot must estimate its location efficiently. What make this goal difficult are ambiguous and noisy measurements alongside the robot interactions with a dynamic environment. With respect to the complexity, the localization problem is categorized in three different problems. In the simplest case the robot knows his initial position [1, 2]. Therefore, the only thing that must be done is tracking the position. This problem is known as the local localization. In a more difficult case that is called global localization, the initial position of the robot is not specified and robot is located in a vague location. Finally the most complex localization problem is the kidnap problem in which the robot may be teleported without its knowledge and must be recovered from this situation as quickly as possible.

The environment of a soccer playing robot is a dynamic and symmetric field. There are some other robots that may affect on the location of the robot and can corrupt its measurements. Moreover sensor measurements are noisy and the robot can be teleported to other locations by external agents. With respect to the mentioned issues, determining the robot pose in a unique point is infeasible. Therefore, the belief of the robot about its position is estimated by probabilistic methods. In term of probability, the localization problem is stated as keeping the probabilistic belief as acquiring noisy measurements. Bayes filter is the basis of all common localization methods. This filter is a recursive model estimating the posterior probability distribution over all possible states due to the collected measurements so far [3]. In favor of the posterior representation, Bayes filter can be implemented in two different approaches: continuous and discrete [3]. In most continoius approaches, the belief is represented by the parameters of a normal distrubution efficiently. these continious methods are unimodal, so thay are only applicable to local localization problems. Extended Kalman Filter (EKF) and Unsecnted Kalman Filter (UKF) are the most populare continious methods. On the other hand, discrete approaches are non-parametric filters that represent the belife with discretizing the state space. Their advantages lie in the ability to estimate arbitary distribution and consequently deal with all localization problems. Monte Carlo Localization (MCL) is the most widely discrete approach implemented in the soccer playing robots.

Combining the efficincy of continious approach with the ability of solving kidnap in discrete ones can lead to a valuable localization method. In this paper, we propose a new hybrid localization method based on a population of UKF and MCL models, utilizaing advantage of both methods while overcoming their weaknesses. Efficiency of our approach is evaluated in various experiments.

II. Related Works

Triangulation is the most trivial localization method that calculates a single point that is closest to the current location of the robot using geometric relations and absolute measurements [2]. But in dynamic systems like soccer playing robots, due to the existence of noise and uncertainty in sensors and environment, the position of the robot cannot be determined uniquely. Hence several probabilistic and fuzzy approaches have been introduced over the last years.

Kalman filter [3] and its extensions are among the most used estimators in state tracking. Kalman filter is a mathematical tool that tracks the state of a dynamic linear system optimally. Linearity condition rarely exists in the robotic systems. To deal with this problem, Extended Kalman Filter and Unscented Kalman Filter are developed. They overcome this problem by linearizing non-linear systems. The EKF uses Taylor expansion to linearize the system. This type of linearization is subject to the approximation error and amount of it depends on the complexity of the functions that have been approximated and the degree of normal variable uncertainty [1]. A superior method that decreases the effect of linearization is UKF [4]. It approximates normal variable with a set of points called sigma points. The sigma points are propagated through the non-linear functions and the parameters of the normal variable are retrieved from the transformed sigma points. The time complexity of UKF is more than EKF with a constant magnitude. Both EKF and UKF are unimodal and can be used only in local localization problems. This issue is addressed in [5] by utilizing a set of UKFs. In this method the location of a robot is estimated with a set of weighted UKFs. The weight of each UKF is smoothly updated according to the absolute measurements. To recover from a probable kidnap, new low weighted UKFs are injected with respect to the last observations. To manage the size of the population the low weighted UKFs are removed. A major drawback of this model arises when the number of positions that are obtained from sensor model is high. In this case, to estimate true posterior, many UKFs are needed that is critical in real-time soccer playing robots with low processing capability.

Another popular filter that is used for dealing with kidnaped and global robot localization problems is MCL [6]. MCL is a non-parametric filter that estimates the belief of the robot position with a finite set of weighted samples. There are several variants of this filter. The basic MCL filter called Sampling Importance Resampling (SIR) has a resampling step in which each sample is duplicated proportional to its weight. SIR suffers from kidnapping, because after some iterations, all of the samples are converged and in this situation there is no chance to retrieve the true position if the robot is teleported. In [7] this problem is addressed with generating a fixed percentage of samples with respect to the absolute measurements in the resampling step. Their proposed method is known as Sensor Resetting Localization (SRL). A more formal method similar to the SRL is presented in [8]. In contrast it weights the new samples from relative measurements. Actually it is a more accurate version of Bayes filter implementation that especially makes robustness in state estimation for highly accurate sensors.

Since the absolute measurements are noisy, adding a fixed part of samples from these measurements can decrease the robustness of estimation. In [9], a new method is proposed for weighting each sample. Instead of changing the weight instantly, it is increased or decreased smoothly. It is clear that using this strategy reduces the sensitivity to the sensor noise. However, using this method makes independent different object observation correlated. To reduce the incorrect correlation among observations, a temporal smoothing MCL (TSMCL) is proposed in [10] that fades out memory of past measurements in the sample weights. Moreover, they suggest a new resampling algorithm called lazy resampling that limits the number of samples duplicated from a particle.

The efficiency of MCL based methods is related to the sample size. A good idea is to have large samples in initial state of localization when there is no prior information about the location. However, in tracking state when the samples density is converged, there is no such a requirement. The Kulback Leiber Distance (KLD) sampling [11] adapts the samples size during the localization process. It chooses the number of samples using the uncertainty of samples density. More formally, KLD sampling determines the number of samples such that the distance between the true posterior and sample-based approximation doesn't exceed a pre-defined threshold. Here, the true posterior is the samples generated after incorporating last relative measurement.

Combining advantages of several localization methods can lead to a more efficient method. In [12] a hybrid localization method relying on a 2D grid-based Markov Localization (ML) and MCL is introduced. The ML represents the state space with a 2D grid and uses histogram filter to approximate the posterior [1]. They named their method Reverse MCL (RMCL). As the name of RMCL implies, it reverses the MCL process, i.e. instead of distributing samples uniformly in the state space and then converging them; RMCL first finds where the location of samples should be converged using coarse grained ML and then samples are drawn there. In the other words, RMCL satisfies global and kidnaped localization problems with a 2D grid and satisfies position tracking with MCL. When the MCL filter is initialized with the ML, this filter is deactivated until the quality of the MCL filter is decreased. A similar approach has been developed in [13]. In contrast it uses an EKF for position tracking and both ML and EKF are executed simultaneously. The key component of their method is a controller that filters the mismatched observations and initializes the EKF in kidnap situations. If the controller mistakes reinitializing EKF due the outliers, then it can make the position estimation unstable. In [14] this problem is solved with using a population of EKFs. Instead of reinitializing old EKF in other locations a new EKF is created. If the new filter has been created because of the sensor noise, the uncertainty of this filter is increased and after some iteration it is removed, while the old ones estimate the true location. Also a combination of grid-based Fuzzy-Markov method and a population of EKFs is established in [15]. It is identical with previous mentioned methods in which the Fuzzy-Markov controls the creation and removing of EKFs. All of these combined methods use the grid based localization to initialize a tracker filter like EKF or MCL and recovering from kidnap. However, the fine-grained grid localization suffers from high processing time overhead.

III. HYBRID LOCALIZATION

As discussed in previous section, Kalman filter extensions such as EKF and UKF have some unique advantages that make them suitable in state tracking. They assume that the distribution of the robot location is a normal variable, therefore, they estimate the location with high efficiency. Moreover, they are light weight in time complexity. The major problem of these filters is that they are only strong for local localization problems. On the other hand, MCL based methods can deal with all localization problems. The weakness of these filters is that achieving high accuracy has much time complexity that is a big issue for soccer playing robots with low resources. In addition, most of them suffer from sensor noises when using small set of particles. So they are unstable while stability in localization makes stability in behavioral unit of the robot. In this section we propose a hybrid localization method that utilizes the advantages of both MCL and UKF while overcomes their weaknesses.

The main goal of this paper is introducing an accurate and multimodal localization method with low processing time overhead. In this section we individually evaluate each MCL and UKF filters on the case of humanoid soccer robot and then present the hybrid localization method in details. The proposed method can be used in every other type of robots employing proper motion and sensor models.

A. Monte Carlo Localization

The MCL represents the belief X_t about robot pose with a finite set of M weighted particles (samples) at time t:

$$X_t = p_t^1, p_t^2, \dots, p_t^M \tag{1}$$

Each particle $p_t^i = (x\ y\ \theta)^T$ is a hypothesis about a possible location of the robot in the field. Here $(x\ y)$ are the Cartesian coordinates and θ is the orientation of the sample in the field of play. Also each sample has a weight w_t^i that shows the likelihood of observing last absolute measurement from the sample position. Initially the samples are distributed uniformly in the whole state space. If there is a prior knowledge about the robot position, the distribution of samples can be done accordingly. Then the samples are updated with the incoming measurements. This process can be stated in three steps: predication, weighting and resampling.

In prediction step we incorporate last relative measurement u_t in sample set using the motion model:

$$p_t^i = p_{t-1}^i + \begin{pmatrix} u_{t,x}\,\mathrm{R}_{1,1}(p_{t-1,\theta}^i) + u_{t,y}\,\mathrm{R}_{1,2}(p_{t-1,\theta}^i) \\ u_{t,x}\,\mathrm{R}_{2,1}(p_{t-1,\theta}^i) + u_{t,y}\,\mathrm{R}_{2,2}(p_{t-1,\theta}^i) \\ u_{t,\theta} \end{pmatrix} + \varepsilon_t \tag{2}$$

Where $u_{t,x}$, $u_{t,y}$ and $u_{t,\theta}$ are the translational and rotational displacements respectively acquired from odometry. Also ε_t is a normal random variable that reflects the noise of the odometry data and $\mathrm{R}(.)$ is a rotation function defined as:

$$R(\varphi) = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix} \tag{3}$$

To weight the samples we incorporate absolute measurements. Here absolute measurements gathered from a camera. The captured image can contains landmarks of the filed that are placed in some known positions. Each detected landmark is a member of percept classes Γ (eg. Goalpost, filed line and etc.). For every sample p_t^i and landmark $j \in \Gamma$ we maintain the filtered importance weight $w_t^{i,j}$. This weight is updated smoothly with respect to the detected landmarks as presented in [10]. Assuming conditional independence of the percept classes the sample weight w_t^i is calculated as:

$$w_t^i = \prod_j w_t^{i,j} \tag{4}$$

Finally in resampling step the samples are duplicated with respect to their weights. Here we employed low variance resampling [1]. To accelerate the convergence of the samples, new samples are generated by absolute measurements when the average weight of samples becomes less than a threshold.

B. Unscented Kalman Filter

Applying non-linear functions on a normal distribution can lead to a non-normal distribution. The UKF is a unimodal filter that represents the state space with a normal variable and solves this issue by using $2L+1$ carefully chosen sigma points. Here $L=3$ is the dimension of the state space. These points capture the true mean and covariance of a multivariate normal distribution. Then the points are propagated through the non-linear functions. Finally the parameter of the desired distribution can be retrieved with high accuracy from theses points.

Assume $X_t \sim \mathcal{N}(s_t, \mathrm{P}_t)$ be the normal variable that we want to estimate. The $s_t = (x\ y\ \theta)^T$ is the mean vector of the normal variable, representing estimated pose of the robot and P_t determines the uncertainty of the state variable. Using this normal variable the sigma points can be generated as follows:

$$sp_i = \begin{Bmatrix} s_t & i = 0 \\ s_t + \sqrt{(L+\lambda)P_t} & i = 1, \dots, L \\ s_t - \sqrt{(L+\lambda)P_t} & i = L+1, \dots, 2L \end{Bmatrix} \tag{5}$$

The $\lambda = \alpha^2(L+\kappa) - L$ is the scaling parameter. α and κ are tuning parameters. The square root of P_t can be calculated by choolsky decomposition.

There are two functions that affect the belief X_t: predict function and correct function. The predict function transforms the X_{t-1} to X_t with respect to the last relative measurement that is gathered from the odometry. The correct function incorporates absolute measurements in X_t to refine it. To apply predict function on X_{t-1}, we have done the following steps:

1. Generate sigma points.
2. Transform every sp_i using predict function.
3. Calculate prior parameters of the normal state variable.

The transformation of the sigma point sp_i is done by:

$$sp_i = sp_i + \begin{pmatrix} u_{t,x}\,\mathrm{R}_{1,1}(sp_{i,\theta}) + u_{t,y}\,\mathrm{R}_{1,2}(sp_{i,\theta}) \\ u_{t,x}\,\mathrm{R}_{2,1}(sp_{i,\theta}) + u_{t,y}\,\mathrm{R}_{2,2}(sp_{i,\theta}) \\ u_{t,\theta} \end{pmatrix} \tag{6}$$

Then new parameters are estimated as follows:

$$\begin{aligned} s_t &= \textstyle\sum_{i=0}^{2L} w_m^i sp_i \\ P_t &= \textstyle\sum_{i=0}^{2L} w_c^i (sp_i - s_t)(sp_i - s_t)^T + R_t \\ w_m^0 &= \tfrac{\lambda}{L+\lambda} \\ w_c^0 &= \tfrac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \\ w_m^i &= w_c^i = \tfrac{1}{2(L+\lambda)}, \qquad i = 1,2,\dots,2L \end{aligned} \tag{7}$$

Where R_t is the noise covariance matrix of the odometry data. Also β is the non-negative weighting parameter and can be used to incorporate prior knowledge of X_t. w_m^i and w_c^i are the weights that specify the participants of each sigma point in the estimated mean and covariance of the normal variable, respectively.

To correct the estimated parameters we use the absolute measurements. It contains landmarks information. Each detected landmark l described by the vector $z_t^l = (\Delta x\ \Delta y)^T$ that shows the x/y distances to the landmark l. For each landmark detected at time t we follow these steps:

1. Generate sigma points from prior parameters.
2. Calculate predicted $\bar{z}_t^{l,i}$ for each sp_iusing correct function.
3. Calculate the Kalman gain.
4. Calculate posterior parameters of the normal variable.

The predicted $\bar{z}_t^{l,i}$ for each sigma point sp_i is computed as:

$$\begin{aligned} dx &= x_l - sp_{i,x} \\ dy &= y_l - sp_{i,y} \\ \bar{z}_t^{l,i} &= \begin{pmatrix} \mathrm{dx}\, \mathrm{R}_{1,1}(sp_{i,\theta}) + dy\, \mathrm{R}_{1,2}(sp_{i,\theta}) \\ dx\, \mathrm{R}_{2,1}(sp_{i,\theta}) + dy\, \mathrm{R}_{2,2}(sp_{i,\theta}) \end{pmatrix} \end{aligned} \tag{8}$$

Where x_l and y_l are the Cartesian coordinates of the landmark l in the field.

The Kalman gain determines how much utilize the landmark l to correct the desired parameters and is measured by the following equation:

$$\begin{aligned} \hat{z}_t^l &= \sum_{i=0}^{2L} w_m^i \bar{z}_t^{l,i} \\ P_{z_t^l, z_t^l} &= \sum_{i=0}^{2L} w_c^i (\bar{z}_t^{l,i} - \hat{z}_t^l)(\bar{z}_t^{l,i} - \hat{z}_t^l)^T + Q_t \\ P_{sp, z_t^l} &= \sum_{i=0}^{2L} w_c^i (sp_i - s_t)(\bar{z}_t^{l,i} - \hat{z}_t^l)^T \\ K_t &= P_{sp, z_t^l}\, P_{z_t^l, z_t^l}{}^{-1} \end{aligned} \tag{9}$$

Where Q_t is the noise covariance matrix of the landmark l and K_t is the desired gain. Finally the posterior parameters are calculated:

$$\begin{aligned} s_t &= s_t + K_t(z_t^l - \hat{z}_t^l) \\ P_t &= P_t - K_t P_{z_t^l, z_t^l} K_t{}^T \end{aligned} \tag{10}$$

If there is more than one correspondence for a detected landmark in the field (like line corners) the correction is accomplished only for the ones with maximum likelihood.

C. combining MCL and UKF

The key idea of this paper is that kidnap and global localization problems can be handled by MCL as quickly as possible and the position tracking can be done with UKF models efficiently. When the uncertainty of the belief is high or a kidnap is detected, the belief is represented by MCL samples to discover the most probable positions. When the samples are converged in the small number of clusters, they are efficiently estimated with UKF models. The overall process of our method is presented in algorithm 1. Initially the samples of MCL are distributed uniformly in the state space (if we haven't any prior information) and there isn't any UKF models (lines 1-3). Line 6 updates the MCL samples with measurements. Lines 7-16 manage the creation of new UKF models. If the MCL samples are converged to a number of pre-defined clusters, they can be replaced with UKF models. For every valid cluster if the number of active UKF models does not exceed a maximum threshold, a UKF hypothesis is replaced by it. The validity of a cluster is determined by the average weight of its samples. To calculate the parameters of the new UKF we have used Maximum Likelihood Estimation of the samples of corresponding cluster. The efficiency of the proposed method is depended to the clustering algorithm. Because it isn't a trivial task and can take a huge time. In [16] a clustering algorithm is introduced that uses the intrinsic features of MCL and clusters the samples in linear time. We have used their method in our algorithm. Lines 19-21 incorporate measurements in UKF models. To control the population of UKF models, low-weighted models are eliminated and similar ones are merged in line 22 and 23, respectively. Similarity of two UKF models i and j are defined with a simplified Mahalanobis distance metric [17]:

Algorithm 1: Hybrid Localization

```
1.  Init_MCL();
2.  is_mcl_actived = true;
3.  ukf_models = ∅;
4.  while true do
5.      if is_mcl_actived then
6.          Update_MCL();
7.          mcl_clusters = Get_MCL_Clusters();
8.          If Count( mcl_clusters) < thredshold_Clusters then
9.              foreach cluster_i in mcl_clusters do
10.                 if Count(ukf_models) < thredshold_ukfs then
11.                     If Validity(cluster_i) > thredshold_validity and
12.                         Replace_New_UKF(ukf_models, cluster_i);
13.                         is_mcl_actived = false;
14.                     end if
15.                 end if
16.             end for
17.         end if
18.     endif
19.     foreach ukf_i in ukf_models do
20.         Update (ukf_i)
21.     end for
22.     Remove_Waste_UKFs();
23.     Merge_UKFs();
24.     if Weight_Best_UKF() < thredshold_reset and not
        (is_mcl_actived) then
25.         Init_MCL();
26.     end if
27. end while
```

$$Dis_{i,j} = (w_i w_j)(s_t^i - s_t^j)^T (w_i D_i + w_j D_j)(s_t^i - s_t^j) \tag{11}$$

Where w_i is the weight of the model i and D_i is a matrix formed by diagonal component of the covariance matrix of model i. To avoid computationally expensive merging algorithm, the parameters of new model are equal to the parameters of the model with more weight. Also The new weight is the sum of w_i and w_j.

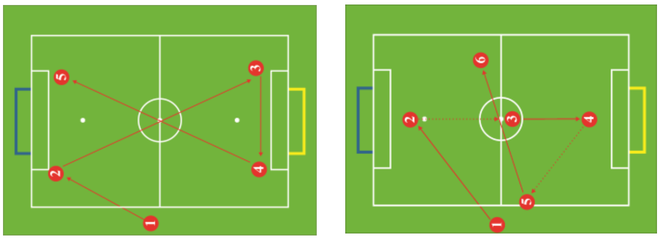

Fig.1. Paths followed in the first and second experiments. The numbers indicate the order.

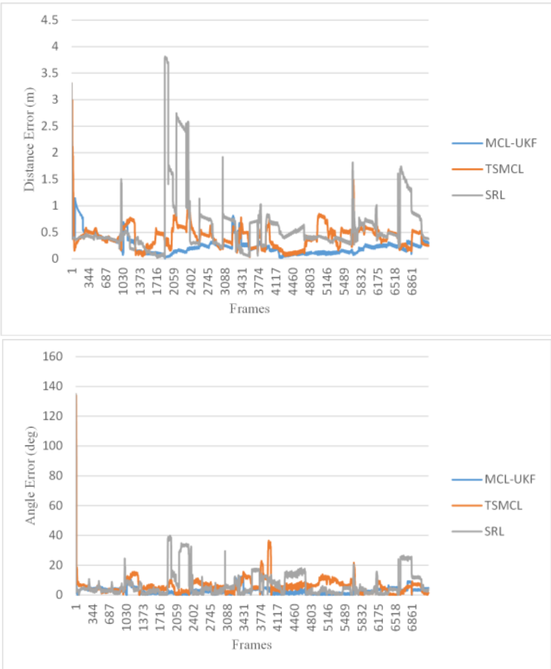

Fig. 2. Error in one trial of the first experiment. Distance Error (top) and Angel Error (bottom).

Line 24-26 are responsible for recovering from a kidnap. We have associated a weight to each UKF. This weight is updated smoothly analogous to a sample of MCL. If the weight of the best UKF (a model with maximum weight) is lower than a threshold, there is the probability that a kidnap has been occurred. If so, MCL samples are reinitialized.

IV. Experiments

The proposed method is implemented on Darwin-op humanoid robot which is used by many research teams around the world and also in RoboCup competitions. To access ground truth information the experiments are performed on the webots simulator [18]. The simulated process runs on a pc equipped with Intel Core 2Due T9550 2.66 GHz processor and 4GB DDR2 Ram. The environment is a 9×6 Robocup humanoid soccer field. Here we haven't concerned about field symmetry. So the color of goals are distinguishable. In all experiments the only landmark used for localization are the goal posts. For all methods we have used 200 samples.

To analyze the efficiency of the proposed hybrid localization (MCL-UKF) we have done some experiments that evaluate the accuracy, relocalization ability and run time complexity. Our method is compared against TSMCL the state of the art particle filter and SRL. All methods are executed simultaneously, so they incorporate the same measurements.

TABLE I. Average of Distance and Angle Errors in the First Experiment

Method	Distance (m)			Angle (deg)		
	Mean	*Median*	*SD*	*Mean*	*Median*	*SD*
MCL-UKF	0.258	0.208	0.309	3.007	2.018	7.045
TSMCL	0.406	0.386	0.235	5.812	4.273	6.489
SRL	0.644	0.474	0.587	10.56	4.449	14.026

TABLE II. Average of Distance and Angle Errors in the Second Experiment

Method	Distance (m)			Angle (deg)		
	Mean	*Median*	*SD*	*Mean*	*Median*	*SD*
MCL-UKF	0.507	0.263	0.848	9.026	3.053	25.541
TSMCL	0.576	0.373	0.699	10.546	4.779	17.685
SRL	0.615	0.398	0.744	9.176	3.454	17.664

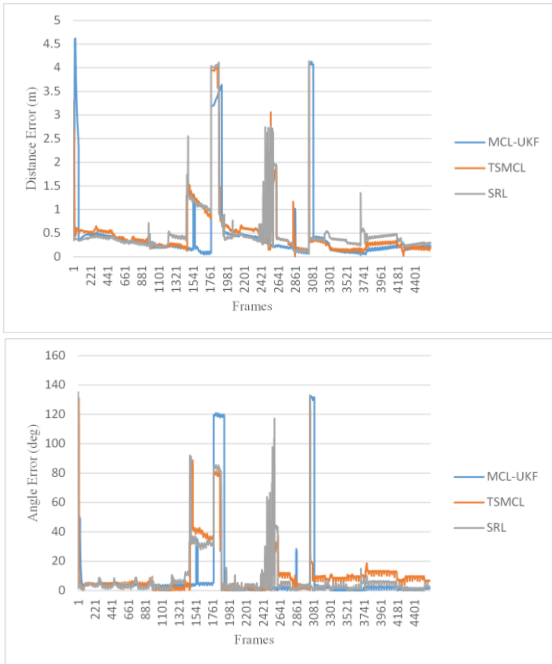

Fig. 3. Error in the second experiment. Distance Error (top) and Angle Error (bottom).

In the first experiment the robot is placed on the side line, near the middle of the field. Initially it hasn't any knowledge about its position. The robot traverses the path illustrated in Fig.1, left. This experiment is repeated for 10 times. The results of one trial of these experiments are plotted in Fig. 2. As presented, all algorithms have converged to the near of the true position after a few updates. However, MCL-UKF outperforms the TSMCL and SRL in tracking the position. It is more accurate and has a better stability on noisy measurements. The error in position and orientation of all trials of this experiment have been summarized in Table I.

The second experiment shows the ability to recovering from kidnap, where the robot is moved suddenly. The path used for this experiment is illustrated in Fig. 1 right. The robot

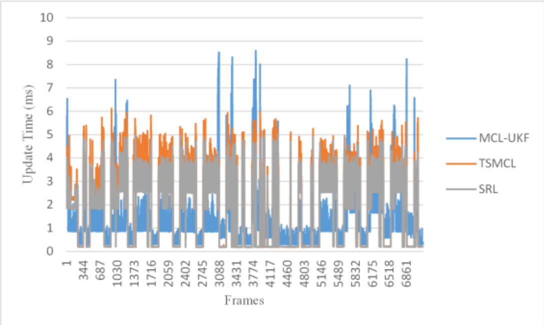

Fig. 4. CPU time of the one trial of the first experiment.

TABLE III. AVERAGE OF UPDATE TIME IN THE FIRST EXPERIMENT

Method	Mean	Median	SD	Min	Max
MCL-UKF	0.992	0.805	0.985	0.29	10.428
TSMCL	1.567	0.222	1.638	0.182	8.552
SRL	1.393	0.219	1.446	0.181	8.521

starts from point one, goes to point two. At this point it is teleported to point three manually and this scenario is repeated from the point four to five. This experiment was repeated 10 times. The results of one trial of these experiments are depicted in Fig. 3. The two kidnaps are occurred in frames 1776 and 3050, respectively. According to the results, all methods have recovered from the kidnaps successfully. However, MCL-UKF and TSMCL retrieved the true location slower than SRL due to the smooth weighting. The result of all trials of these experiments have been summarized in Table II.

In the last experiment the processing time of the methods are tested. Time complexity is critical for a soccer playing robot. A time consuming localization method can affect other functional unit of the robot. To analyze the CPU time we have used the update time of the executions of the first experiment. The details of the processing time in each frame for one trial of the experiment are demonstrated in fig. 4. Also the results of update time are summarized in Table III. The CPU performance of our method is improved about 30% and 37% compared to SRL and TSMCL, respectively. The maximum update time in a frame is recorded for MCL-UKF. In some frames, some UKF models and MCL samples are updated at the same time that leads to increase in update time. However, this situation occurs rarely and in more than 90% of the frames the average update time is in the interval (0, 2.5) while this is (0, 4.5) and (0, 4) for TSMCL and SRL, respectively.

V. CONCLUTION

In this paper a new hybrid localization method based on MCL and UKF has been presented. The proposed method combines the advantages of both algorithms to reach an accurate, stable and real-time localization method that can deal with all types of localization problems. The global and kidnaped problems are managed by initializing MCL samples. While the accurate and robust position tracking are accomplished by UKF models. These UKF Hypothesis are created with respect to the state of the MCL samples such that, whenever they are converged to some predefined clusters new UKF models are initialized in the position of the valid clusters. The efficiency of the hybrid method is evaluated by some experiments and the results show the effectiveness of this method. The proposed method is implemented on real robot and also tested on the simulator for easy access to ground truth model.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge: MITPress, 2005.

[2] H. Kose, B. Celik, and H. L. Akın, "Comparison of Localization Methods for a Robot Soccer Team," *International Journal of Advanced Robotic Systems*, vol. 3, pp. 295-302, 2006.

[3] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Trans. of the ASME, Journal of Basic Engineering*, vol. 82, pp. 34-45, (1960)

[4] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. of 11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, pp.182 -193 1997.

[5] G. Jochmann, S. Kerner, S. Tasse, and O. Urbann, “Efficient Multi-hypotheses Unscented Kalman Filtering for Robust Localization,” in *RoboCup 2011: Robot Soccer World Cup XV* SE - 19, vol. 7416, T. Röfer, N. M. Mayer, J. Savage, and U. Saranlı, Eds. Springer Berlin Heidelberg, 2012, pp. 222–233.

[6] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte-Carlo localization for mobile robots,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1999.

[7] S. Lenser and M. Veloso. “Sensor resetting localization for poorly modelled mobile robots,” in *The Int. Conf. on Robotics and Automation*, April 2000.

[8] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. “Robust Monte Carlolocalization for mobile robots,” *Journal of Artificial Intelligence*, 2001.

[9] T. Rofer and M. Jungel, “Vision-based fast and reactive Monte Carlo localization,” in *The IEEE Int. Conf. on Robotics and Automation*, pages 856-861, Taipei, Taiwan, 2003.

[10] W. Nisticò and M. Hebbel, “Particle Filter with Temporal Smoothing for Mobile Robot Vision-Based Localization,” in *Informatics in Control, Automation and Robotics* SE - 12, vol. 37, J. Cetto, J.-L. Ferrier, and J. Filipe, Eds. Springer Berlin Heidelberg, 2009, pp. 167–180.

[11] D. Fox, "Adapting the sample size in particle filters through KLD-sampling," *Int. J. Robotics Res. (IJRR)*, vol. 22, no. 12, pp.985 -1004 2003.

[12] H. Köse and H. L. Akın, “The Reverse Monte Carlo localization algorithm,” Rob. Auton. Syst., vol. 55, no. 6, pp. 480–489, Jun. 2007.

[13] J. S. Gutmann "Markov-Kalman localization for mobile robots," in *Proc. of 16th Int. Conf. Pattern Recog.*, vol. 2, pp.601 -604 2002.

[14] F. Mart´ın, C. Ag¨uero, and J. e. M. Ca˜nas, "Extended Kalman filter populations for a reliable real-time robot self-localization," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2012.

[15] F. Martın, V. Matellan, P. Barrera, and J. M.Canas, "Localization of legged robots combining a fuzzy-Markov method and a population of extended Kalman filters," *Robotics and Autonomous Systems*, vol. 55, pp. 870–880, 2007.

[16] T. Laue and T. Röfer, "Pose Extraction from Sample Sets in Robot Self-Localization - A Comparison and a Novel Approach," in *4th European Conf. on Mobile Robots*, Mlini/Dubrovnik, Croatia, 2009, pp. 283–28.

[17] M. Quinlan and R. Middleton, "Multiple Model Kalman Filters: A Localization Technique for RoboCup Soccer," in *RoboCup 2009: Robot Soccer World Cup XIII*. vol. 5949, J. Baltes, M. Lagoudakis, T. Naruse, and S. Ghidary, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 276-287.

[18] O. Michel, "Webots: Professional Mobile Robot Simulation," *International Journal of Advanced Robotic Systems*, vol. 1, pp. 39-42, 2004.