

# A Comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM Algorithms

Zeyneb Kurt-Yavuz and Sirma Yavuz

Computer Engineering Department, Yildiz Technical University, Istanbul, Turkey  
zeyneb@ce.yildiz.edu.tr, sirma@ce.yildiz.edu.tr

**Abstract**—This study aims to contribute a comparison of various simultaneous localization and mapping (SLAM) algorithms that have been proposed in literature. The performance of Extended Kalman Filter (EKF) SLAM, Unscented Kalman Filter (UKF) SLAM, EKF-based FastSLAM version 2.0, and UKF-based FastSLAM (uFastSLAM) algorithms are compared in terms of accuracy of state estimations for localization of a robot and mapping of its environment. The algorithms were run using the same type of robot on Player/Stage environment. The results show that the UKF-based FastSLAM has the best performance in terms of accuracy of localization and mapping. Unlike most of the previous applications of FastSLAM in literature, no waypoints are used in this study.

## I. INTRODUCTION

The simultaneous localization and map building (SLAM) techniques tries to solve the problem where an autonomous vehicle starts in an unknown location in an unknown environment to incrementally build a map of this environment. The robot uses this map to compute its own location simultaneously. Uncertainty is what makes the SLAM problem hard to solve. The robot can only make noisy, probabilistic observations of its surroundings without knowing its exact location. Every time the robot moves uncertainty is added into an already uncertain pose.

Among the methods proposed in literature to solve the SLAM problem, the methods based on Bayesian estimation theory has been the most successful ones.

Many successful applications exist in literature using EKF to solve nonlinear estimation problems as position tracking, localization and SLAM [1-10] problems. But the quadratic computational complexity of the EKF makes it difficult to apply in real time. UKF is a more reliable estimator than EKF while the system model is highly nonlinear. The past of the UKF is relatively short compared to EKF. By approximating the probability density function instead of the nonlinear function itself, UKF SLAM [11,12] received a considerable attention. Yet it did not make any improvement to the computational complexity of the EKF. FastSLAM [13-17] utilizes particle filters and improves the computational complexity considerably compared to EKF and UKF. UKF-based FastSLAM (uFastSLAM), the newest approach proposed in literature [18-23], combines the more reliable estimation ability of UKF and reduced computational complexity feature of FastSLAM.

In this study, the localization and mapping accuracy of EKF, UKF, FastSLAM version 2.0 and UKF-based FastSLAM (uFastSLAM) were compared by using several particle numbers in FastSLAM and uFastSLAM to achieve more reliable observation about those algorithms. Furthermore, the effects of angular velocity noise changing on the root mean square errors (RMSE) of the robot position, obtained by those algorithms, were compared.

Between the second and sixth section, Bayes Filter, EKF, UKF, FastSLAM 2.0 and uFastSLAM are described, respectively. Section seven illustrates the experimental results; conclusion is given in the final section.

## II. BAYES FILTERS

All methods realized in this study are based on Bayes Filters. The state of the system, the state of the robot and the environment, at time  $t$  is expressed by random variables  $x_t$ . In probabilistic SLAM methods, the state of the robot and the environment can only be expressed through the conditional probability distributions of the sensor data which are available for the robot.

The probability distribution representing the uncertainty at each point in time is called *belief*,  $bel(x_t)$ . Bayes filters apply two update rules successively to predict the system state.

The predictive belief at time  $t$  is calculated just before the observation ( $z_t$ ) and uses the control data ( $u_{1:t}$ ) by the time  $t$ . This step is also called control update, and expressed as in (1).

$$\overline{bel}(x_t) = P(x_t | z_{1:t-1}, u_{1:t}) \quad (1)$$

The state estimate given in (1) is *corrected* according to (2), using sensor measurements ( $z_{1:t}$ ) and the control data ( $u_{1:t}$ ) by the time  $t$ . This step is called as measurement update or the posterior belief of the system state and calculated whenever a sensor provides a new observation.

$$bel(x_t) = P(x_t | z_{1:t}, u_{1:t}) \quad (2)$$

---

This research has been supported by Yildiz Technical University Scientific Research Projects Coordination Department. Project Number: 2010-01-02-DOP01.

The system is assumed to be Markov, so the observations and controls are conditionally independent of past measurements and control readings, given the true state.

The SLAM algorithms presented in this study strongly differ in how they represent these probability densities and the belief  $bel(x_t)$ . These differences of the algorithms are briefly explained in following sections.

### III. EKF SLAM

Kalman filters are the most widely used variant of Bayes filters. Standard Kalman filter assumes linear state transitions and linear measurement transitions with added Gaussian noise.

The extended Kalman filter (EKF) overcomes the linearity assumption by linearizing the nonlinear state and measurement transition functions via Taylor expansion.

The nonlinear state transition function  $g$  and measurement transition function  $h$  is expressed as in (3) and (4) respectively with added noises  $\varepsilon_t$  and  $\delta_t$ .

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \quad (3)$$

$$z_t = h(x_t) + \delta_t \quad (4)$$

The extended Kalman filter represents beliefs by the mean vector  $\mu_t$  and the covariance matrix  $\Sigma_t$  at time  $t$ .

The Taylor expansion of state transition function  $g$  is given as in (5) and (6). The Jacobian  $G_t$  is the value of first derivative of  $g$  at the point  $\mu_{t-1}$ .

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + g'(u_t, \mu_{t-1})(x_{t-1} - \mu_{t-1}) \quad (5)$$

$$g(u_t, x_{t-1}) = g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}) \quad (6)$$

Similarly, the Taylor expansion of measurement transition function  $h$  is given as in (7) and (8). The Jacobian  $H_t$  is the value of first derivative of  $g$  at the point  $\mu_{t-1}$ .

$$h(x_t) \approx h(\bar{\mu}_t) + h'(\bar{\mu}_t)(x_t - \bar{\mu}_t) \quad (7)$$

$$h(x_t) = h(\bar{\mu}_t) + H_t(x_t - \bar{\mu}_t) \quad (8)$$

The update rules corresponding to the prediction and correction steps of the Bayes filter and other details of the EKF algorithm can be found in [8-10].

### IV. UKF SLAM

The unscented Kalman filter (UKF) is a direct application of the unscented transform. Unscented transform is based on the idea that it is easier to approximate the probability function instead of a nonlinear function [11]. The method first used in SLAM problem by Martinez-Cantin and Castellanos [12]

Instead of approximating the state and measurement transition functions by Taylor series expansion, the UKF extracts  $2n+1$  sigma points,  $X^{[i]}$ , from the Gaussian as in (9) and passes these through the nonlinear state and measurement functions.  $n$  is the dimension of feature state,  $\lambda$  is computed as in (10) where  $\alpha$  and  $k$  scaling parameters determine how far the sigma points are separated from the mean.

$$\begin{aligned} X^{[0]} &= \mu \\ X^{[i]} &= \mu + (\sqrt{(n+\lambda)\Sigma})_i, \quad \text{for } i=1, \dots, n \\ X^{[i]} &= \mu - (\sqrt{(n+\lambda)\Sigma})_{i-n}, \quad \text{for } i=n+1, \dots, 2n \end{aligned} \quad (9)$$

$$\lambda = \alpha^2 (n+k) - n \quad (10)$$

$\mu$  is the mean vector in previous step,  $\chi[i]$  is the *sigma points* on Cartesian coordinate and  $\Sigma$  is the covariance matrix of previous step. After the computation, each sigma point is transformed via the nonlinear state transition function  $g$  as in (11).

$$y^{[i]} = g(X^{[i]}) \quad (11)$$

Then, the mean vector and the covariance matrix are predicted by multiplying the extracted sigma points and their weights. Predicted measurements are obtained by the measurement transition function  $h$  as given in (12).

$$\bar{z}_t^{[i]} = h(X^{[i]}) \quad (12)$$

Posterior estimation steps of the mean vector and the covariance matrix can be found in [8,11,12]. Unlike EKF, UKF does not employ a linearization process via Taylor expansion which causes incomplete representation of the nonlinear functions and does not employ Jacobian matrices for calculating feature covariance [18].

### V. EKF-BASED FASTSLAM

The FastSLAM algorithm uses particle filters to estimate robot position and uses EKF to estimate landmark positions [8, 13-17]. FastSLAM algorithm maintains a set of particles, each of these particles has its own belief regarding positions of the robot and  $N$  landmarks. These beliefs are the local Gaussians and each particle uses EKF for predictions and updates of the landmark positions. Each one of the  $M$  particles in the system has the form as given in (13). The notation  $[m]$  is the index of the particle while  $x^{t,[m]} = (x \ y \ \theta)^T$  represents particle's path

estimate and  $\mu_{i,t}^{[m]}$  and  $\Sigma_{i,t}^{[m]}$  are the mean vector and covariance matrix of the Gaussian representing the  $i$ th landmark.

$$X_t^{[m]} = x_t^{[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \dots, \mu_{i,t}^{[m]}, \Sigma_{i,t}^{[m]}, \dots, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \quad (13)$$

In FastSLAM 1.0, the control  $u_t$  is used to sample new robot pose for each particle according to the motion model. FastSLAM 2.0 uses the measurement  $z_t$  and the control  $u_t$  together to sample new robot pose, therefore FastSLAM 2.0 is more efficient than FastSLAM 1.0. However, its implementation is more difficult than the implementation of FastSLAM 1.0. Details can be found in [8, 13-17].

If a landmark  $m_n$  is observed again, EKF measurement update equations are used for updating the landmark estimations. FastSLAM linearizes the measurement model in the same way as EKF does. Temporary particle set, containing  $M$  particles, is resampled according to an importance factor. Stratified resampling is used in this study [24].

EKF and UKF performs only one data association hypothesis over all state space, however in the FastSLAM each particle has its own hypothesis for this problem. FastSLAM provides several local solutions for localization and mapping by using the particles.

## VI. UKF-BASED FASTSLAM

There are number of FastSLAM and UKF based particle filter applications[13-23,25]. Generally, they use the simulation provided by Bailey *et. al.*[26]. This simulation assumes an array of waypoints with known coordinates and the control signals are generated to direct the robot from one waypoint to another. Utilization of waypoints prevents these applications to be used in completely unknown environments like search and rescue sites. Such waypoints are not used in this study.

In UKF-based FastSLAM, UKF measurement update equations take place instead of EKF measurement update operations of FastSLAM 2.0. UKF-based FastSLAM equations is out of scope of this paper, the details can be found in [23] widely.

## VII. EXPERIMENTAL RESULTS

In this section the tools used for the experiments are explained and the results are given in a classified manner.

### A. Experimental Setup

Algorithms presented in this study are coded in C++ to be used on mobile robot developed by our team. The initial results presented here are obtained in Player/Stage environment [27]. Player is a network server used for robot control, control algorithms connect to the server as clients to send control signals to the robot or to receive sensor data from the robot.

Kinematic models are based on a three-wheeled robot as shown in Fig. 1. The proximity data are provided by a laser range finder placed on top of the robot and the odometer data are provided by a wheel encoder. A simple wall-following method as presented in [28] is used for exploration.

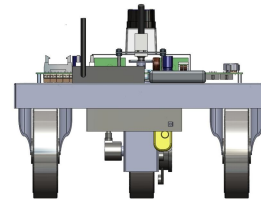


Figure 1. Three-wheeled robot model used for kinematic models.

Three different environment setup, shown in Fig. 2 were used to evaluate the performance of different SLAM techniques.

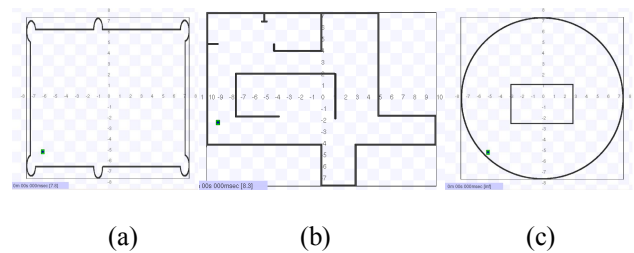


Figure 2. Test environments used for mapping

In following sections, firstly, the resulting maps for all three environments are given. After that, the effect of particle numbers in FastSLAM 2.0 and uFastSLAM algorithms on the RMS error of robot position estimates under same velocity and noise parameters are analyzed. In the same figure RMSEs of EKF and UKF algorithms are also given. Then, the position and orientation errors of the algorithms according to the loop size (travelled distance in terms of meters) are given, when the particle number of FastSLAM 2.0 and uFastSLAM is 50. Finally RMS position errors of the algorithms according to changing of angular velocity noise parameter is presented.

To obtain the results, described above, the average RMSE values of each algorithm were acquired by running each of the algorithms 10 times.

### B. Maps Generated by the Algorithms

Maps generated by the EKF, UKF, FastSLAM 2.0, and uFastSLAM algorithms for the first test environment are shown in Figs. 3, 4, 5, and 6 respectively. EKF predicts the robot path more faulty than the other methods, as it is examined in following sections total position errors achieved by the algorithm are not the smallest and shows high deviations from the ground truth. In the map generated by UKF, it can be seen that the accuracy of the map is bigger than the EKF and FastSLAM2.0 (when particle number is 50). uFastSLAM gives the best accuracy results for localization and mapping.

the uFastSLAM is visibly more accurate than the other algorithms.

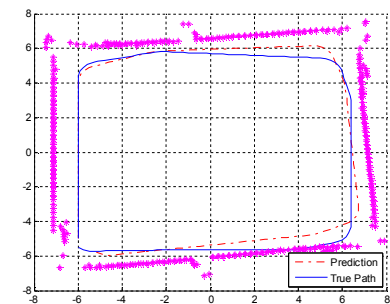


Figure 3. Map of the environment (a) generated by EKF SLAM.

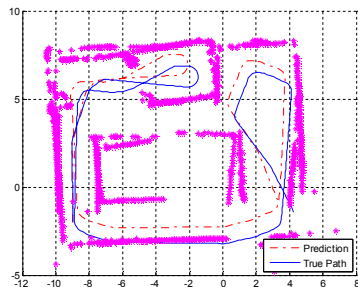


Figure 7. Map of the environment (b) generated by EKF SLAM.

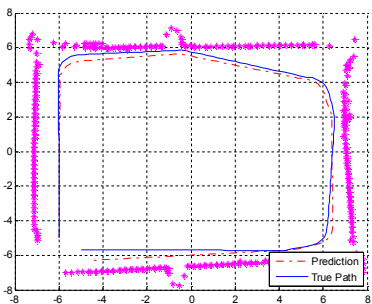


Figure 4. Map of the environment (a) generated by UKF SLAM.

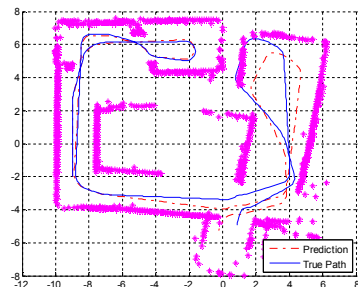


Figure 8. Map of the environment (b) generated by UKF SLAM.

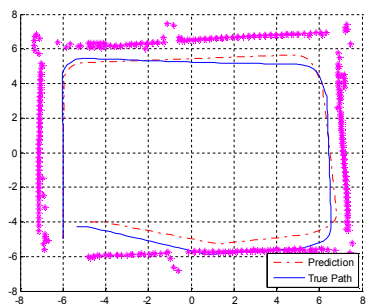


Figure 5. Map of the environment (a) generated by FastSLAM2.0.

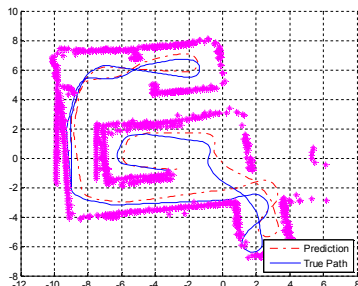


Figure 9. Map of the environment (b) generated by FastSLAM2.0.

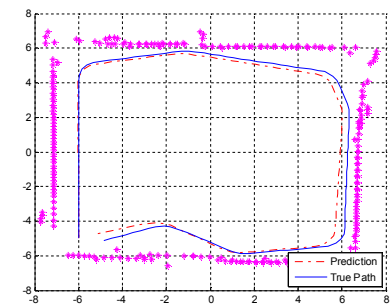


Figure 6. Map of the environment (a) generated by uFastSLAM.

Figs. 7, 8, 9, and 10 show the maps generated for the second test environment. Results show that the performance of

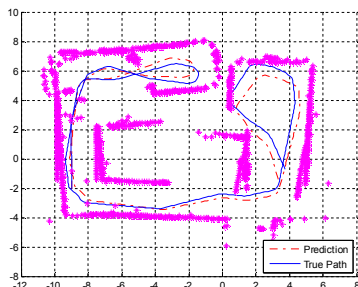


Figure 10. Map of the environment (b) generated by uFastSLAM

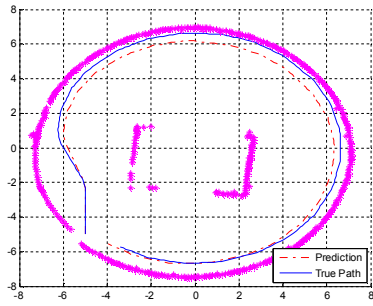


Figure 11. Map of the environment (c) generated by EKF SLAM.

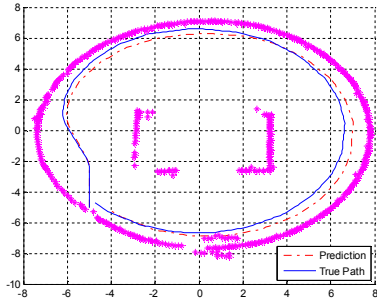


Figure 12. Map of the environment (c) generated by UKF SLAM.

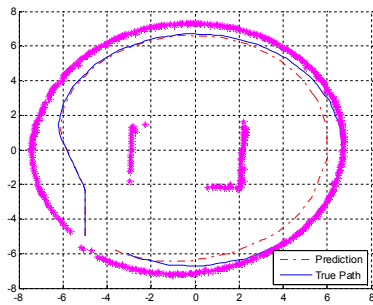


Figure 13. Map of the environment (c) generated by FastSLAM2.0.

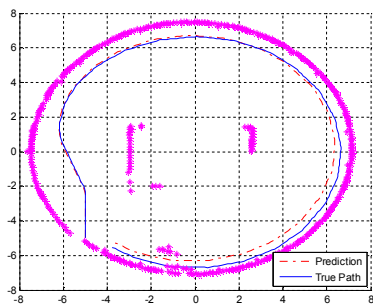


Figure 14. Map of the environment (c) generated by uFastSLAM.

For the third circular environment, Figs. 11, 12, 13, and 14 show the maps generated by the algorithms. The map generated by the uFastSLAM algorithm is successful visually, than the other algorithms in circular formed environment.

C. RMS Robot Position Errors respect to Particle Numbers

In this section average RMS errors of estimated robot positions are examined. Each algorithm was run 10 times and average RMSE of position was calculated. Number of particles was chosen as 1, 5, 10, 50, 100, and 500. Translational velocity is 0.3m/sec; std. dev. of the translational and rotational velocities are 0.01m/sec and 0.005 rad/sec respectively.

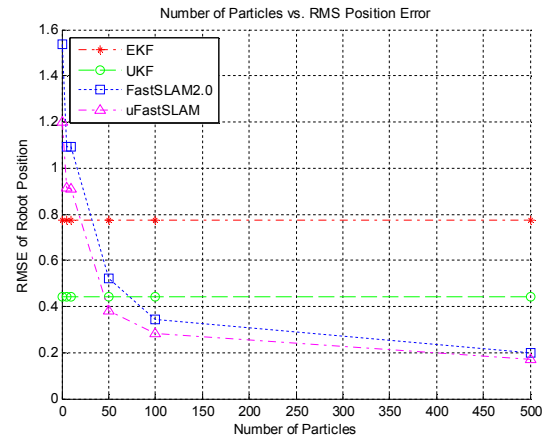


Figure 15. The effect of the particle numbers on the RMSE of robot positions.

TABLE I.  
ESTIMATED ROBOT POSITION RMS ERRORS OF THE ALGORITHMS

Algorithm	RMSE	Algorithm	RMSE
FastSLAM, $M=1$	1.536	uFastSLAM, $M=1$	1.201
FastSLAM, $M=5$	1.094	uFastSLAM, $M=5$	0.916
FastSLAM, $M=10$	1.093	uFastSLAM, $M=10$	0.912
FastSLAM, $M=50$	0.521	uFastSLAM, $M=50$	0.382
FastSLAM, $M=100$	0.342	uFastSLAM, $M=100$	0.285
FastSLAM, $M=500$	0.199	uFastSLAM, $M=500$	0.169
EKF	0.775	UKF	0.441

Fig. 15 shows the RMS errors of estimated robot positions for all four algorithms. Since the number of particles ( $M$ ) affects the performance of the FastSLAM algorithms, results are given for different number of particles for those algorithms. The noise parameters are same for all experiments. The effect of the particle number on the RMSE is also given in Table I. Table I and Fig. 15 clearly illustrate that, up to a particular particle number, FastSLAM algorithms cannot accomplish better results than EKF and UKF. FastSLAM2.0 outperform EKF and UKF when the particle number is bigger than 50; however uFastSLAM outperform the other three algorithms when the particle number is equal and bigger than 50. Thus the

most efficient algorithm is uFastSLAM when the particle number is chosen appropriately.

*D. RMS Robot Position and Orientation Errors*

During the runtime of the algorithms position and orientation estimation errors with respect to the travelled distance (in terms of m.) are given in Figs. 16 and 17 respectively.

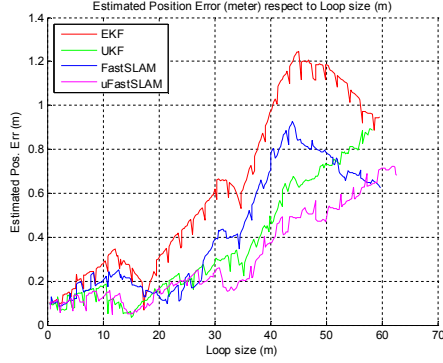


Figure 16. Position estimation errors of four algorithms during the runtime.

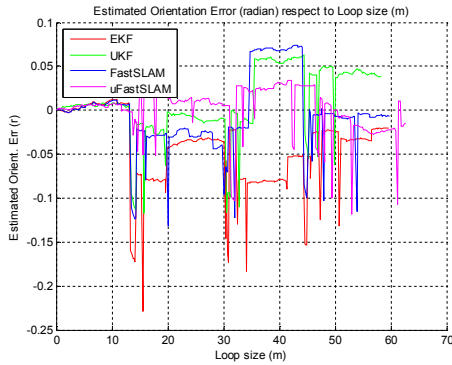


Figure 17. Orientation estimation errors of four algorithms during the runtime

Figs. 16 and 17 clearly indicate that the minimum position and orientation estimation error is achieved by uFastSLAM, while the particle number is 50. UKF is the second efficient algorithm, since its position and orientation error is less than EKF and FastSLAM 2.0 while the number of particles is 50.

*E. RMS Robot Position Errors according to Rotational Velocity Noise Parameter*

The last comparison was made on all four algorithms which were run under different amounts of rotational velocity noise and the RMS errors of each are given in Table II and Fig. 18. Table II depicts that; uFastSLAM is significantly the most reliable algorithm in case of rotational velocity noise changing. UKF is more reliable and efficient than the EKF and FastSLAM 2.0. EKF is the least efficient one among them.

Maximum likelihood unknown data association was applied in all of the experiments for each algorithm. The second environment in Fig. 2(b), that is the most complicated one, was used in the experimental result sections C, D, E.

TABLE II.  
RMS ERRORS OF POSITION ESTIMATIONS

Std. of the Angular Noise	RMSE of Position Estimation			
	EKF	UKF	FastSLAM (M=50)	uFastSLAM (M=50)
0,02	1.563	1.114	1.225	1.038
0,04	2.389	1.371	2.364	1.218
0,06	3.004	1.687	2.578	1.529
0,08	3.347	1.962	2.635	1.878
0,10	3.556	2.263	2.665	2.082
0,12	3.796	2.362	2.711	2.174
0,14	3.830	2.709	2.854	2.203
0,16	4.225	2.920	3.177	2.418
0,18	4.5253	2.732	3.772	3.021
0,20	4.807	4.102	4.328	3.580

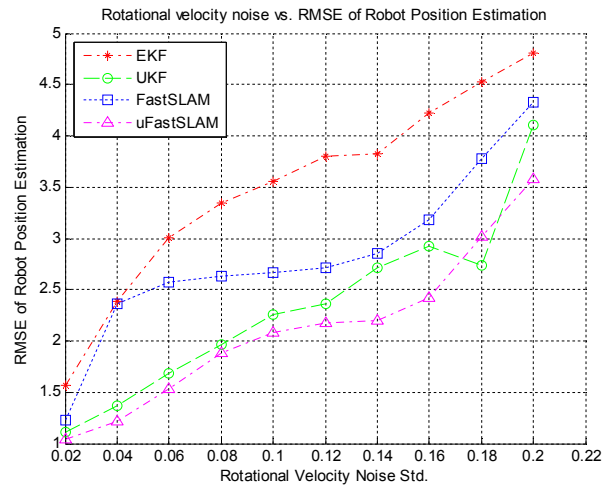


Figure 18. RMS errors of robot positions under different rotational noises

VIII. CONCLUSION

Although EKF remains to be a popular choice for the solution of SLAM problem, as a result of the experiments in this study, UKF-based FastSLAM is observed to be the most efficient algorithm among standard EKF, UKF, FastSLAM 2.0, and uFastSLAM. Our results also show that EKF-based FastSLAM can perform as well as UKF SLAM algorithm, while the particle number is equal and greater than 100, or EKF algorithm itself. Theoretical complexities of the EKF and UKF are same but in application the selection and computation of the sigma points caused UKF algorithm to be a slow algorithm compared to the EKF algorithm. uFastSLAM is the slowest one; however it is the most accurate one among them. Also in case of angular velocity noise's increasing, uFastSLAM gives the least RMSE value for robot position estimation.

## REFERENCES

- [1] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman, "A Framework for Vision Based Bearing Only 3D SLAM", IEEE International Conference on Robotics and Automation (ICRA), pp. 1944-1950, 2006.
- [2] T. Lemaire and S. Lacroix, (2007), "SLAM with Panoramic Vision", Journal of Field Robotics, Vol. 24, no. 1-2, pp. 91 - 111, Feb 2007.
- [3] J.-H. Kim, and M.J. Chung, "SLAM with omni-directional stereo vision sensor" In Proceedings of the International Conference on Intelligent Robots and Systems, Vol. 51, no. 1, pp. 31-43, January, 2008.
- [4] J. Artieda, J. M. Sebastian, P. Campoy, J. F. Correa, I. F. Mondragón, C. Martínez and M. Olivares, "Visual 3-D SLAM from UAVs", Journal of Intelligent and Robotic Systems, Vol. 55 (4-5), pp. 299-321, Aug 2009.
- [5] D. Scaramuzza, R. Siegwart and A. Martinelli, (2009), "A Robust Descriptor for Tracking Vertical Lines in Omnidirectional Images and Its Use in Mobile Robotics", International Journal of Robotics Research, Vol. 28, no. 2, pp. 149-171, Feb 2009.
- [6] J. Sola, A. Monin, M. Devy and T. Vidal-Calleja, "Fusing Monocular Information in Multicamera SLAM", IEEE Transactions on Robotics, Vol. 24, no. 5, pp. 958-968, 2008.
- [7] A.P. Gee, D. Chekhlov, A. Calway and W. Mayol-Cuevas, "Discovering Higher Level Structure in Visual SLAM", IEEE Transactions on Robotics, Vol. 24, no. 5, pp. 980–990, October 2008.
- [8] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, The MIT Press, 2005.
- [9] P. Abbeel, A. Coates, M. Montemerlo, A. Ng, and S. Thrun, "Discriminative training of Kalman filters". In Proceedings of Robotics Science and Systems, Cambridge, MA, 2005. MIT Press.
- [10] Q.-H. Meng, Y.-C. Sun, and Z.-L. Cao, "Adaptive Extended Kalman Filter (AEKF)-based Mobile Robot Localization Using Sonar", Journal of Robotica, Vol 18(5), pp. 459–473, September 2000.
- [11] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," Proceedings of the IEEE, Vol. 92, No. 3, March 2004, pp.401-422.
- [12] R. Martínez-Cantin and J. A. Castellanos, "Unscented slam for large-scale outdoor environments", IEEE/RSJ Intl. Conf. On Intelligent Robots and Systems, 2005, pp 328-333.
- [13] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem", In the Proceedings of the AAAI National Conference of Artificial Intelligence, Canada, 2002.
- [14] C. Stachniss, D. Hahnel, and W. Burgard, W., "Exploration with active loop-closing for FastSLAM", Proceedings of International Conference on Intelligent Robots and Systems (IROS), pp. 1505-1510, Oct 2004.
- [15] J. Z. Sasiadek, A. Monjzab and D. Neculescu, "Navigation of an autonomous mobile robot using EKF-SLAM and FastSLAM", 16th Mediterranean Conference on Control and Automation, pp. 517 - 522, 25-27 June 2008, Ajaccio, France.
- [16] M. Montemerlo, and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM", in the Proceedings of the 2003 IEEE International Conference on Robotics & Automation (ICRA), pp. 1985 - 1991, 14-19 September 2003, Taipei, Taiwan.
- [17] T. Bailey, J. Nieto and E. Nebot, "Consistency of the FastSLAM algorithm", IEEE International Conference on Robotics and Automation (ICRA), pp. 424 - 429, 15-19 May 2006, Orlando, FL.
- [18] C. Kim, R. Sakhivel and W. K. Chung, "Unscented FastSLAM: A Robust and Efficient Solution to the SLAM Problem", IEEE Transactions on Robotics, Vol. 24, no. 4, August 2008, pp. 808-820.
- [19] M. Cugliari and F. Martinelli, "A FastSLAM algorithm based on the Unscented Filtering with Adaptive Selective Resampling", 6th International Conference on Field and Service Robotics (FSR 2007), France 2007.
- [20] X. Wang and H. Zhang, "A UPF-UKF Framework for SLAM", IEEE International Conference on Robotics and Automation - ICRA 2007, pp. 1664-1669, April 2007, Rome, Italy.
- [21] L. Zhang, X. Meng and Y. Chen, "Unscented Transform for SLAM Using Gaussian Mixture Model with Particle Filter", International Conference on Electronic Computer Technology, pp. 12-17, 20-22 Feb. 2009, Macau.
- [22] S. Zandara and A. Nicholson, "Square Root Unscented Particle Filtering for Grid Mapping", in the Proceedings of the 22nd Australasian Joint Conference on Advances in Artificial Intelligence, 2009.
- [23] A. Sakai, T. Saitoh and Y. Kuroda, "Robust Landmark Estimation and Unscented Particle Sampling for SLAM in Dynamic Outdoor Environment", Journal of Robotics and Mechatronics, pp. 140-149, Vol.22 No.2 Apr. 2010.
- [24] G. Kitagawa. "Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models", J. Comput. Graph.Statist., 1:1–25, 1996.
- [25] R. Merwe, A. Doucet, N. Freitas and E. Wan, "The Unscented Particle Filter", Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000.
- [26] <http://www-personal.acfr.usyd.edu.au/tbailey/software/index.html>
- [27] B. Gerkey, R. T. Vaughan and A. Howard. "The Player/Stage project: tools for multi-robot and distributed sensor systems". In Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), pp. 317-323, June 2003, Coimbra, Portugal.
- [28] K. Beevers, "Loop closing in topological maps", Proceedings of the International Conference on Robotics and Automation, pp. 4378-4383, April 2005.