

## Comparison of the SLAM algorithms: Hangar experiments

Mehmet Korkmaz<sup>1,a</sup>, Nihat Yılmaz<sup>1</sup>, Akif Durdu<sup>1</sup>

<sup>1</sup> Selcuk University in Konya, TURKEY

**Abstract.** This study purposes to compare two known algorithms in an application scenario of simultaneous localization and mapping (SLAM) and to present issues related with them as well. Mostly used SLAM algorithms Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are compared with respect to the point of accuracy of the robot states, localization and mapping. Because of considering the most implementations in the previous studies, the simulation environments are chosen as big as possible to provide reliable results. In this study, two different hangar regions are tried to be simulated. According to the outcomes of the applications, UKF-based SLAM algorithm has superior performance over the EKF-based one, apart from elapsed time.

### 1 Introduction

Autonomous mobile robots should perform the behaviours or any given tasks with a high degree of autonomy. It is vital for guiding the robot in a desired manner. Therefore, a robot has to know answers of two critical questions: “Where am I?”, “What does the world look like?”. The first question addresses to the problem which is known as a localization problem. According to the localization problem, a mobile robot needs to know its position (location, orientation) to achieve the goal [1]. On the other hand, the second question is related with the environmental information. The robot seeks the answer of what is the world around its. This problem is known as a mapping phenomenon, in the literature. In this problem, a robot tries to build a map or a plan of its environment with an acquired sensor data [2].

In addition to the above mentioned parts, it is good to specify that robot positions and environment information are known for mapping and localization problem, respectively. With these thought, simultaneous localization and mapping (SLAM) problem addresses to the problem that lacking of position and surrounding data. Accordingly, SLAM by a mobile robot is concerned with the problem of creating a map of an unknown environment. The robot has its on-board sensors while simultaneously determining its location using the created map. Because of the fact that, truly autonomy system needs to location and position of features of robot and environment, SLAM problem solution is seen a “holy grail” for the mobile robotic community [3]. The background of SLAM problem is based upon the paper of Smith et al. [4] by using stochastically approach to the problem. After that, SLAM problem has attracted distinct attention from the many researches and mobile robotics communities in the past twenty five years [3,5].

From emerging to now SLAM has been found widely usage area. Early studies, still ongoing, of SLAM are performed by using EKF [6], then particle filter based solutions are shown up [7], after that using of the information-state approaches have come up [8]. Kalman filter based solutions have been still remained on the researchers’ agenda. On the other hand, visual SLAM based approaches have been aroused attention from many studies on this field [9].

For the solution of SLAM problem, plenty of techniques has been applied, but among them Bayesian based approaches have founded the most useful and successful ones. According to the Bayesian method, sensor data observed from the robot are expressed thorough a conditional probability distribution. Then, Bayesian filter uses these distributions as an input and performs two phases that are prediction and correction steps to solve the problem. In the prediction phase, robot motion model is taken into consideration and next state of the robot is predicted. After that, related observations are taken by sensors and used to correct the states.

The structure of the paper is as follows: First, the mathematical probabilistic definitions on SLAM problem is described to show the bases of it in section II. Thereafter, in section III, used robot dynamics, which is based upon Ackermann Modell, are shown. Moreover, EKF and UKF based SLAM algorithms’ foundation are elucidated in depth and state equations have been revealed. Finally, implementation of the algorithms is realized using the thought simulation environments and Root Mean Squared Error (RMSE) and true-estimated paths are pointed out.

<sup>a</sup> Corresponding author: mkorkmaz@selcuk.edu.tr

## 2 SLAM problem

SLAM is an exploration of environment while simultaneously tracking the robot's pose. In order to overcome the problem, probabilistic approaches have been applied successively, up to now. SLAM problem is defined in two manners, which are online SLAM (eq. 1) and full SLAM (eq. 2).

$$P(x_k, m | Z_{1:k}, U_{1:k}) \quad (1)$$

$$P(x_{1:k}, m | Z_{1:k}, U_{1:k}) \quad (2)$$

Full SLAM can be distinguished from the online SLAM only if the posterior are calculated over the entire path  $x_{1:k}$  along with the map. In the equations,  $x$  represents the location and orientation of the robot,  $U_k$  is a control vector,  $m$  denotes the  $i^{\text{th}}$  feature location and  $Z_k$  is an observation taken via robot's sensor.

As a result of Bayes' theorem and assumptions that the robot motion model is Markov and the landmarks are time invariant in the environment. With this thought, SLAM formula can be reproduced as in the eq. 3 [10, 3].

$$\eta \cdot P(z_k | x_k, m) \int P(x_k | x_{k-1}, u_k) (x_{k-1}, m | Z_{k-1}, U_{k-1}) dx_{k-1} \quad (3)$$

$\eta$  is a normalizing constant which comes from assumptions stage. From the point of probabilistic view, calculation of the probabilistic equations are in need of motion and observation models that possesses the robot (eq. 4-5). The motion model is obtained from the robot's dynamic equation while the observation one is derived according to the sensor readings.

$$P(x_k | x_{k-1}, U_k) \quad (4)$$

$$P(z_k | x_k, m) \quad (5)$$

The eq. 3 tells that two stages are implemented to find SLAM problem solution: time update (prediction), prior state estimation and robot motion model are combined; measurement update (correction), this calculation is corrected and associated with the sensor observation and finally result is normalized to appropriate interval.

## 3 Robot model and Algorithms

In this section, Ackermann steering geometry that is used for the robot model and EKF-UKF algorithms are explained in detail.

### 3.1 Robot Model

In the study, used robot motion model has Ackermann steering geometry. It is generally accepted having poor kinematic model but also is a good representation for slow motions [11, 12]. Following equations point out the motion and observation models ((6), (7)), respectively.

$$x_k = \begin{bmatrix} x_{v_k} \\ y_{v_k} \\ \phi_{v_k} \end{bmatrix} = \begin{bmatrix} x_{v_{k-1}} + V_k \nabla t \cos(\phi_{v_{k-1}} + \alpha_k) \\ y_{v_{k-1}} + V_k \nabla t \sin(\phi_{v_{k-1}} + \alpha_k) \\ \phi_{v_{k-1}} + V_k \nabla t \sin(\alpha_k) / L \end{bmatrix} \quad (6)$$

$$z_{k,i} = \begin{bmatrix} r_{k,i} \\ \theta_{k,i} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_{k,i} - x_{v_k})^2 + (y_{k,i} - y_{v_k})^2} \\ \tan^{-1} \frac{y_{k,i} - y_{v_k}}{x_{k,i} - x_{v_k}} - \phi_{v_k} \end{bmatrix} \quad (7)$$

The equations above are summarized with the following definitions:  $x_k = [x_{v_k}, y_{v_k}, \phi_{v_k}]^T$  robot pose information,  $m_i = [x_{k,i}, y_{k,i}]^T$  location of the feature which is also called landmark,  $U_k = [V_k, \alpha_k]^T$  control vector  $z_k = [r_k, \theta_k]^T$  observation. Where the equations symbols means are identified as follows:  $V_k$  velocity,  $\alpha_k$  robot steering,  $r_k$  observed range,  $\theta_k$  observed bearing,  $L$  wheel base of the robot,  $\nabla t$  time interval between  $k$  and  $k - 1$ .

The eq. 6 is called velocity motion model and the other one, 7, represents the range-bearing observation model (Fig. 1 and Fig. 2).

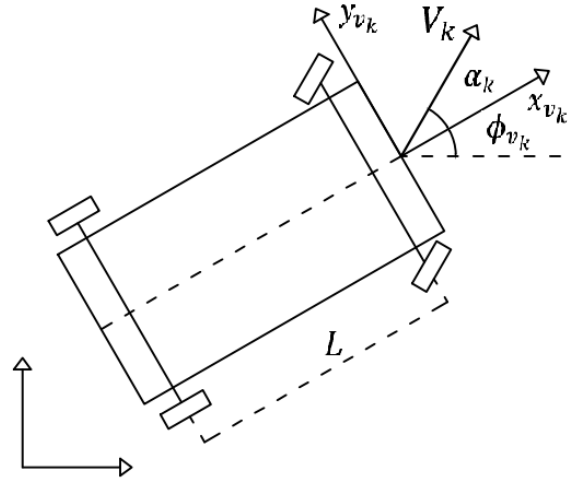


Figure 1. Velocity motion model.

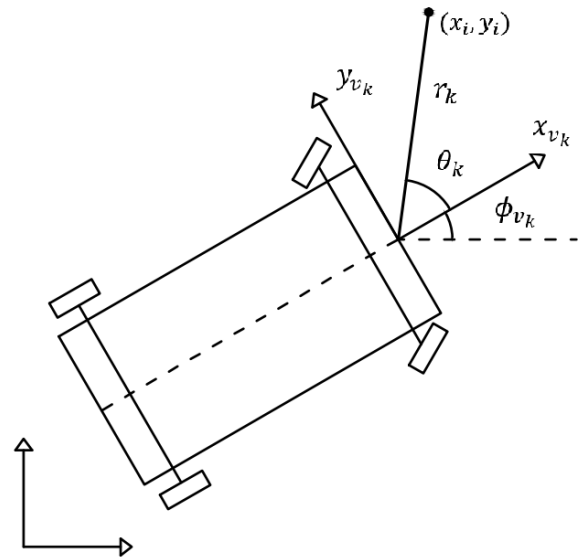


Figure 2. Range-bearing observation model.

### 3.2 EKF-based SLAM

Extended Kalman Filter (EKF) is the oldest and one of the mostly used techniques to implement SLAM problem, in the literature. EKF-based SLAM algorithm is a result of Bayesian Filter and extension of Kalman Filter (KF) for the form of nonlinear cases. The main goal of the EKF is to find posterior distribution of the mean and covariance ((8), (9)) by applying the prediction and correction steps [13, 3, 14].

$$\bar{x} = \mu_k = \begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_{k|k} \end{bmatrix} \quad (8)$$

$$P_{k|k} = \begin{bmatrix} P_{xx} & P_{xm} \\ P_{xm}^T & P_{mm} \end{bmatrix}_{k|k} \quad (9)$$

In the time update or prediction step, robot motion model and robot's mean-covariance are used to predict the mean and covariance, at time k-1. At this step, used robot motion model ( $P(x_k|x_{k-1}, U_k)$ ) is described in the eq. 10.

$$x_k = f(x_{k-1}, U_k) + w_k \quad (10)$$

In the equation,  $f(\cdot)$  refers to robot's kinematics and  $w_k$  is an additive, zero mean Gaussian disturbance ( $w_k \sim \mathcal{N}(0, Q_k)$ ).

According to this step, predicted mean and covariance are evaluated with the following equations ((11), (12)).

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, U_k) \quad (11)$$

$$P_{xx,k|k-1} = \nabla f P_{xx,k-1|k-1} \nabla f^T + Q_k \quad (12)$$

The most prominent part of the EKF is  $\nabla f$ , which means Jacobian of  $f$  calculated at time k-1.

The second stage is a measurement update or correction that provides with the estimation of the prediction via obtained measurements and measurement model. Measurement model of the robot ( $P(z_k|x_k, m)$ ) can be thought as an equation in 13.

$$z_x = h(x_k, m) + v_k \quad (13)$$

In the equation  $h(\cdot)$  denotes the geometry of the measurement, which is similar to motion model, and  $v_k$  is an additive, zero mean Gaussian disturbance ( $v_k \sim \mathcal{N}(0, R_k)$ ). Predicted measurement are calculated from the (13) after that the important points that are innovation and its covariance are obtained with respect to real measurements and predicted values ((14), (15)).

$$V_k = z_k - h(\hat{x}_{k|k-1}, \hat{m}_{k-1}) \quad (14)$$

$$S_k = \nabla h P_{k|k-1} \nabla h^T + R_k \quad (15)$$

As in the  $\nabla f$ ,  $\nabla h$  is a Jacobian of the  $h$  evaluated at preceding step. After from all these steps, estimation is finished ((16), (17)), states are corrected and if necessary next state can be repeated using the same steps.

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_{k|k} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{m}_{k|k-1} \end{bmatrix} + K_k V_k \quad (16)$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \quad (17)$$

$K$  is a Kalman gain and calculated as the following (18).

$$K_k = P_{k|k-1} \nabla h^T S_k^{-1} \quad (18)$$

### 3.3 UKF-based SLAM

Unscented Kalman Filter (UKF) is an alternative way to implement SLAM problem. UKF is mainly based on unscented transform. The main idea on the UKF is to express the probability distribution with random variables, so called sigma points. Instead of using nonlinear function to define probability distribution by Taylor expansion, it is chosen  $2n + 1$  sigma points from the Gaussian (19).

$$\begin{aligned} \chi^{[0]} &= \bar{x} = \mu \\ \chi^{[i]} &= \bar{x} + \left( \sqrt{(n + \lambda)P} \right)_i, \text{ for } i = 1, \dots, n \\ \chi^{[i]} &= \bar{x} - \left( \sqrt{(n + \lambda)P} \right)_i, \text{ for } i = n + 1, \dots, 2n \end{aligned} \quad (19)$$

where  $\lambda$  is defined,

$$\lambda = \alpha^2(n + k) - n \quad (20)$$

$\alpha$  (usually set to  $1e - 4 \leq \alpha \leq 1$ ) and  $k$  (generally chosen 0) are the scaling factors that show the spreads of sigma points from the mean.

In the light of unscented transform information, mentioned sigma points are replacement in the motion model (21) and predicted mean and covariance are calculated according to the equations (22), (23), respectively.

$$\chi_{k|k-1}^i = f(\chi_{k-1|k-1}^i, U_k) \quad (21)$$

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2n} w_i \chi_{k|k-1}^i \quad (22)$$

$$P_{k|k-1} = \sum_{i=0}^{2n} w_i [\chi_{k|k-1}^i - \hat{x}_{k|k-1}][\chi_{k|k-1}^i - \hat{x}_{k|k-1}]^T + Q_k \quad (23)$$

In the correction step, sigma points are substituted in the measurement (observation) model ((24), (25)).

$$z_{k|k-1}^i = h(\chi_{k|k-1}^i) \quad (24)$$

$$\hat{z}_{k|k-1} = \sum_{i=0}^{2n} w_i z_{k|k-1}^i \quad (25)$$

$$V_k = z_k - \hat{z}_{k|k-1} \quad (26)$$

$$S_k = \sum_{i=0}^{2n} w_i [z_{k|k-1}^i - \hat{z}_{k|k-1}][z_{k|k-1}^i - \hat{z}_{k|k-1}]^T + R_k \quad (27)$$

(26) is called innovation and (27) is an innovation covariance. After the measurement, corrected mean and covariance at time k-1 are evaluated with respect to (16) and (17) as in the same EKF-based SLAM. In the above equations,  $K$  is a Kalman gain (28) and is  $P_{xz}$  cross correlation matrix (29).

$$K_k = P_{xz,k|k-1} S_k^{-1} \quad (28)$$

$$P_{xz,k|k-1} = \sum_{i=0}^{2n} w_i [\chi_{k|k-1}^i - \hat{x}_{k|k-1}][z_{k|k-1}^i - \hat{z}_{k|k-1}]^T \quad (29)$$

As it is seen in the formulas UKF-based SLAM differs from the EKF one that is not include Jacobian calculation which generally provides an advantage of easy calculation [15, 16].

### 4 Simulation results

The proposed SLAM algorithms both EKF and UKF are simulated using the MATLAB program. In the simulation, the robot dynamic is thought as Ackermann model. The robot wheel base is set to 4 meters, speed is 3 m/s. In addition to this, sensor maximum distance is accepted as a 30 meters and time interval,  $\nabla t$ , between k and k-1 is supposed to be 0.025 seconds.

In the study, two different scenarios are implemented that are hangar\_1 (square low) and hangar\_2 (cycloid). The simulated maps are indicated in the figures, 3 and 4. According to the simulation, hangar environments are nearly chosen in the size of 200 by 200 meters to see the effectiveness of the algorithms in a big size region. It is approximately placed a special feature, called also landmark, in each ten meters.

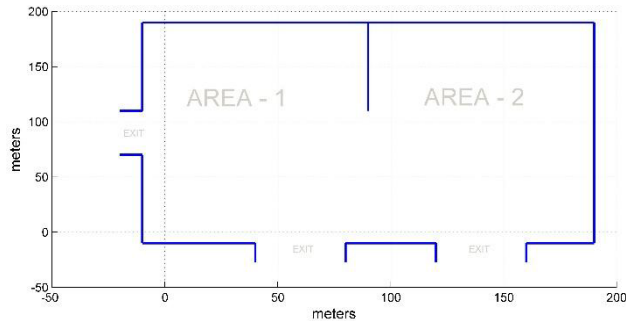


Figure 3. Hangar\_1 (square low) simulation environment

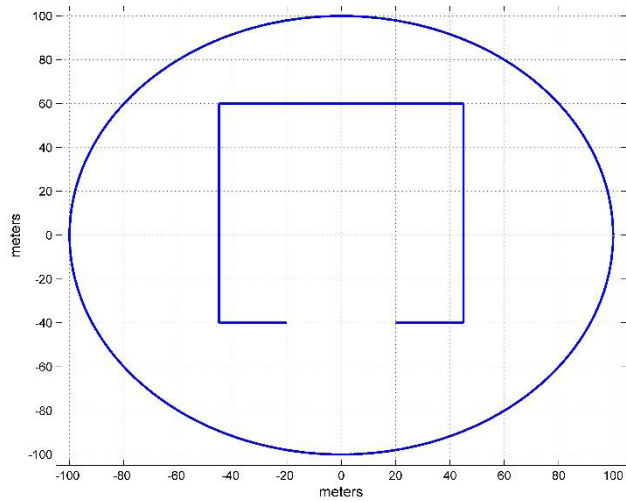


Figure 4. Hangar\_2 (cycloid) simulation environment

It is used a root mean square error (RMSE) criteria to see the productivity of the algorithms for both position (30) and heading angle, subtracting from true data to estimated one.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}{N}} \quad (30)$$

where,  $x_i$  and  $y_i$  are a true paths and  $\hat{x}_i$  and  $\hat{y}_i$  are an estimated paths by algorithms, N is an number of state obtained in simulation.

For the fair comparison, each algorithm is run ten times and average RMSE criteria results are indicated in the table 1 and 2, for each experiment.

Table 1. Hangar\_1 (square low) RMSE and Time Results.

Algorithm	Position error	Heading error	Time (s)
EKF	4,43432	0,78978	62,1163
UKF	2,55331	0,60402	72,1636

Table 2. Hangar\_2 (cycloid) RMSE and Time Results.

Algorithm	Position error	Heading error	Time (s)
EKF	3,63777	0,54790	46,0049
UKF	1,78697	0,20794	49,6039

Figure 5 and 6 are the results of the algorithms' estimated path versus true robot path. In addition to previous results these figures also show the supremacy of the UKF-based algorithm, which is the closest one to the true path.

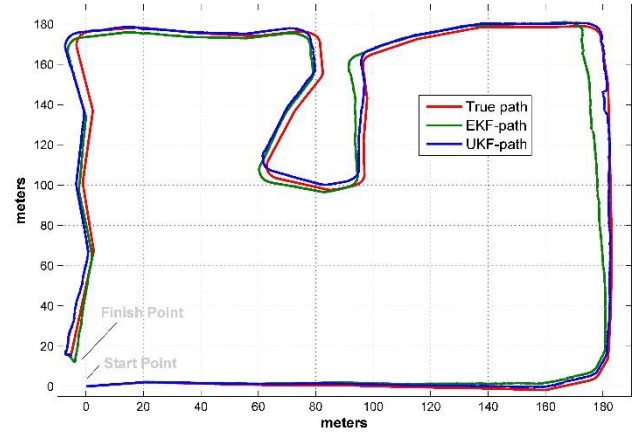


Figure 5. Hangar\_1 (square low) true path vs algorithm's estimated path

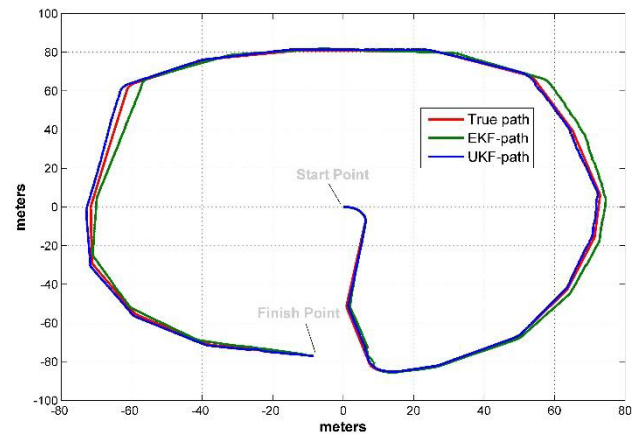


Figure 6. Hangar\_2 (cycloid) true path vs algorithm's estimated path

## 5 Conclusion

EKF is one of the best and classical algorithm to the solution of SLAM problem. Although its easy implementation and effectiveness are verified various studies, new solution to SLAM problem are required. Besides this, UKF is one of the mostly used techniques and powerful solution to the SLAM problem. In this study, both EKF and UKF algorithms are applied to the big size environment to see the effectuality of the algorithms. First of all, simulation environments are generated taking consideration to the square low and cyclical hangar plans. After that, landmarks are established nearly each ten meters and robot waypoints are determined to optimal vision of sensor. Finally, produced data files are loaded to the workspace and algorithms are tested navigating the robot through waypoints. Both position error and heading error evaluation results show the superiority of the UKF to EKF (table 1, 2). While the RMSE of the robot pose is superior from the point of the UKF, elapsed time to complete the route is not as good as in EKF.

## Acknowledgements

This work is supported by Selcuk University Scientific Research Fund, BAP. Authors are thankful to RAC-LAB (<http://www.rac-lab.com/>) for providing the trial version of their commercial software.

## References

1. Z. Liang, X. Ma, & X. Dai, Extended Monte Carlo algorithm to collaborate distributed sensors for mobile robot localization. In *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on* (pp. 1647-1652), IEEE, (2007, December)
2. S. Thrun, Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 1-35, (2002)
3. H. Durrant-Whyte, & T. Bailey, Simultaneous localization and mapping: part I, *IEEE Robot. Autom. Mag.* **13(2)**, 99-110, (2006)
4. R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *International Symposium of Robotics Research*, pages 467-474, (1987)
5. T. Bailey, J. Nieto, J. Guivant, M. Stevens, & E. Nebot, Consistency of the EKF-SLAM algorithm. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (pp. 3562-3568). IEEE, (2006, October)
6. R. Smith and P. Cheesman, "On the representation of spatial uncertainty," *Int. J. Robot. Res.*, **5(4)**, pp. 56-68, (1987)
7. G. Dissanayake, H.F. Durrant-Whyte and Tim Bailey, "A Computationally Efficient Solution to the Simultaneous Localization and Map Building Problem", In *IEEE Int. Conf. on Robotics and Automation*, **2**, pp. 1009-1014, (2000).
8. S. Thrun, Y. Liu, D. Koller, A. Ng, and H. Durrant-Whyte, "Simultaneous localisation and mapping with sparse extended information filters," *Int. J. Robot. Res.*, **23 (7-8)**, pp. 693-716, (2004).
9. J. Fuentes-Pacheco, J. Ruiz-Ascencio, & J.M. Rendón-Mancha, Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, **43(1)**, 55-81, (2015)
10. C.C. Wang, C. Thorpe, & S. Thrun, Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, **1**, pp. 842-849). IEEE, (2003, September)
11. S. Lavelle, *Planning Algorithms*. Cambridge: Cambridge University Press, (2006)
12. L. Marin, M. Vallés, A. Soriano, A. Valera, & P. Albertos, Event-Based Localization in Ackermann Steering Limited Resource Mobile Robots, *IEEE/ASME Trans. Mechatronics*, **19(4)**, 1171-1182 (2014)
13. A. Chatterjee, A. Rakshit, & N.N. Singh, Simultaneous Localization and Mapping (SLAM) in Mobile Robots. In *Vision Based Autonomous Robot Navigation* (pp. 167-206). Springer Berlin Heidelberg, (2013)
14. P.S. Maybeck, *Stochastic Models, Estimation and Control*, vol. I. New York: Academic 1979.
15. S. Li, & P. Ni, Square-root unscented Kalman filter based simultaneous localization and mapping. In *Information and Automation (ICIA), 2010 IEEE International Conference on* (pp. 2384-2388). IEEE, (2010, June)
16. R. Havangi, M.A. Nekoui, H.D. Taghirad, & M. Teshnehlab, SLAM based on intelligent unscented Kalman filter. In *The 2nd International Conference on Control, Instrumentation and Automation*, (2011)