

The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014)

Localization Performance Evaluation of Extended Kalman Filter in Wireless Sensors Network

Rafiullah Khan^{a,*}, Sarmad Ullah Khan^b, Shahid Khan^c and Mohammad Usman Ali Khan^d

^a *Universitat Politècnica de Catalunya (UPC) Barcelona, Spain Email: rafiullah.khan@upc.edu*

^b *CECOS University of IT and Emerging Sciences, Peshawar, KPK, Pakistan Email: sarmad@cecos.edu.pk*

^c *COMSATS Institute of Information Technology, Abbottabad, KPK, Pakistan Email: shahid@ciit.net.pk*

^d *University of Engineering and Technology, Peshawar, KPK, Pakistan Email: musmank@gmail.com*

Abstract

This paper evaluates the positioning and tracking performance of Extended Kalman Filter (EKF) in wireless sensors network. The EKF is a linear approximation of statistical Kalman Filter (KF) and has the capability to work efficiently in non-linear systems. The EKF is based on an iterative process of estimating current state information from the previously estimated state. Its working is based on the linearization of observation model around the mean of current state information. The EKF has small computation complexity and requires low memory compared to other Bayesian algorithms which makes it very suitable for low powered mobile devices. This paper evaluates the localization and tracking performance of EKF for (i) Position (P) model, (ii) Position-Velocity (PV) model and (iii) Position-Velocity-Acceleration (PVA) model. The EKF processes distance measurements from cricket sensors that are acquired through time difference of arrival between ultrasound and Radio Frequency (RF) signals. Further, localization performance under varying number of beacons/sensors is also evaluated in this paper.

© 2014 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Selection and Peer-review under responsibility of the Program Chairs.

Keywords: Wireless Sensors Network; Positioning; Tracking; Extended Kalman Filter

1. Introduction

Fast and accurate localization is very important in many applications of wireless sensor networks (WSNs) including security systems for vehicles or buildings, indoor positioning/navigation, bridges or structures monitoring, health monitoring for industrial machines, military applications etc. To improve positioning/localization accuracy in

* Corresponding author. Tel.: +34-684 076 695.
E-mail address: rafiullah.khan@upc.edu

critical and sensitive applications, different methods have been proposed in literature. There are two important factors for methods classification: (i) type of the range measurements, and (ii) technology used. Some of the most commonly used range measurements in WSN include: (i) angle of arrival (AOA), (ii) direction of arrival (DOA), (iii) received signal strength (RSS), (iv) time difference of arrival (TDOA), and (v) time of arrival (TOA). The generic taxonomy of positioning systems is depicted in Fig. 1. One of the most common technique for localization is trilateration that uses distance measurements obtained through TOA, TDOA or any other method. Fig. 2 below depicts the trilateration scenario. A1, A2 and A3 represent the known location of fixed anchor nodes transmission the signal where r_1 , r_2 and r_3 respectively, represent the distance of each anchor node from the receiver or mobile node. In the absence of noise all the three circles will intersect at a common point. Fig. 2 depicts the noise scenario where distance measurements can fluctuate through a margin represented with red circles. Thus, tracking of mobile node is difficult and can be located at any point in the triangular region formed through the intersection of red circles.

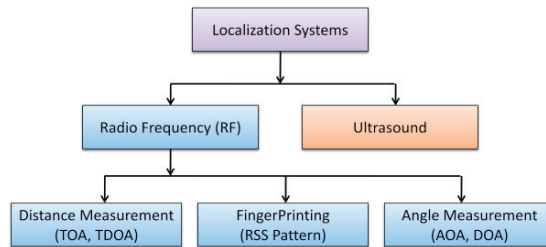


Fig. 1. Taxonomy of localization systems.

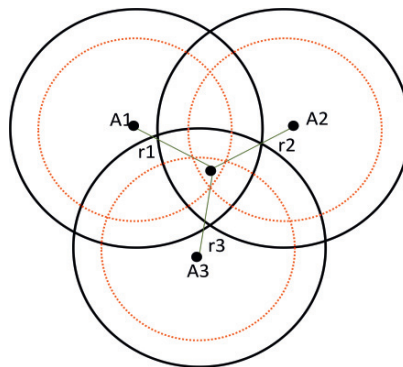


Fig. 2. Trilateration Localization.

A set of range measurements is required from the anchor nodes for position estimation or tracking of the mobile node in WSN. In indoor environment, range measurements are most probable to be affected by errors due to fading, reflections or multipath. Thus, accurate positioning and tracking is quite challenging task in indoor environment. This paper presents positioning and tracking algorithms based on the linear approximation of Kalman Filter (KF) called Extended Kalman Filter (EKF). The EKF can efficiently work in non-linear systems which makes it quite suitable especially for WSN applications. Further, the EKF has very low memory requirement and computation complexity compared to other Bayesian positioning and tracking algorithms such as Particle Filter (PF).

In this paper, we used cricket sensors and evaluated the localization and tracking performance of EKF for (i) Position (P) model, (ii) Position-Velocity (PV) model and (iii) Position-Velocity-Acceleration (PVA) model. The cricket localization scheme is based on the Radio Frequency (RF) and ultrasound technologies for distance estimation. The cricket sensors or beacons were manually placed on the floor that transmit the information on RF signal. The beacons also transmit a concurrent ultrasound signal with each of its RF advertisement. The listener/mobile node measures the time difference of arrival of the two signals to determine its distance from each beacon. The mobile node takes advantage of the speed difference between ultrasound signal (speed of sound) and

RF signal (speed of light) for distance estimation as depicted in Fig. 3.

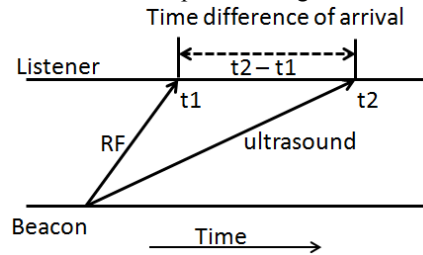


Fig. 3. Time difference of arrival between ultrasound signal (speed of sound) and RF signal (speed of light).

The remaining paper is formatted as follows. Section 2 describes the related previous work from the literature. Section 3 explains the EKF algorithm. Section 4 describes the state equations for 2 dimensional positioning. Section 5 evaluates the localization performance. Finally, section 6 presents the concluding remarks.

2. Related Work

In the literature, different distributed localization algorithms have been proposed. The localization algorithms can be characterized based on whether they depend or not on anchor nodes, where anchor nodes have preconfigured true position. Another way of characterizing localization algorithms is based on whether the algorithm is concurrent or incremental. The anchor-based algorithms use some nodes with known position [1]. The position can be determined by manual settings/configuration or with the help of other localization mechanism. A coordinate system is required to which the positions of anchor nodes are referenced. The anchor-based localization algorithms accuracy usually depends on the number of anchor nodes. Higher number of anchor nodes result in small position error. Anchor-free algorithms on the other hand don't use any preconfigured fixed position nodes and rely on the local distance data to estimate the node position [2]. Incremental positioning algorithms start with three or four nodes with pre-configured position, and then other appropriate nodes are added to this set after calculating their positions using measured distances to the previously pre-computed nodes positions. On the other hand, in concurrent positioning algorithms, all the nodes in parallel calculate their position information using iterative optimization schemes [3].

The EKF algorithm is most widely used for nonlinear systems which linearizes the observation model. The EKF has been extensively used in many areas including probabilistic robotics [4], neural networks [6], image processing or depth recovery [5], and global navigation satellite systems [7]. The EKF is the extended version of Kalman filter to better handle non-linearities. In all applications, the EKF core basically remains same however its computation complexity may vary depending on how state information is described or presented and updated. Some other filters have been proposed in the literature, such as the Central Difference Filter (CDF), Iterated Extended Kalman Filter (IEKF) and Divided Difference Filter [8]. The conventional EKF and IEKF algorithms do not take into account the linearization errors and lead towards inconsistent state estimates. The Unscented Kalman Filter (UKF) is based on the deterministic approach using samples to address this problem. It collects a set of sample points carefully to easily estimate the true mean and covariance. A variation of UKF algorithm has been proposed in [9] which is named Reduced Sigma Point Filter (SPF) but it does not take in consideration the linearization errors.

A Particle Filter (PF) has also been proposed in literature which is based on the idea of representing the posterior density function with a set of particles each with its own associated weight. The PF estimates the position based on the random particles and their associated weights. Comparing PF to KF, the former has a more robust performance for non-Gaussian and non-linear system due to the simulated posterior distribution. The PF algorithm requires more memory and is much more computationally complex compared to the EKF as PF requires very large number of particles to correctly represent the posterior distribution. An Unscented Particle Filter (UPF) has been introduced in [10] that incorporate the latest observation into prior estimates of the PF. The UPF outperforms in terms of convergence compared to PF, EKF and UKF but it has very high computational cost.

3. Positioning Through Extended Kalman Filter

Accurate and fast localization and tracking algorithms have very high computation and memory requirements. The constrained devices such as wireless sensors usually require algorithms with low computational and memory requirements. The Kalman Filter (KF) has initially been proposed for linear systems. It comparatively requires low computation power compared to other Bayesian algorithms and has been extensively studied in the past

few years for its usability in wireless sensors [8], [11]. The main limitation of KF is the requirement of linear state information and estimation equations. Further, the KF requires noise to have Gaussian distribution [11]. Unfortunately, the measurements are usually affected by noise generated from many different sources which may or may not have Gaussian distribution. Thus, the EKF has been recently proposed and studied for non-linear systems which is an efficient approximation of KF linear state model. Previous experiments in literature have concluded that EKF has comparatively very low computational and memory requirements than other Bayesian algorithms, thus making it very suitable for mobile devices.

The EKF is based on iterative process of estimating the current position or state information from the previous state information. The system dynamics are considered very important that determines the EKF overall performance. The state transition model of EKF can be presented in a generalized form as follows:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}, \quad (1)$$

where $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ represents the process noise. The process noise is considered to have normal distribution which has zero mean and covariance matrix \mathbf{Q}_k . The \mathbf{x}_k and \mathbf{x}_{k-1} denote the current and previous posteriori state vectors at time t_k and t_{k-1} , respectively. The f represents a non-linear function which deals with the prediction of current state information from the previous state information. The observation model for EKF can be presented as follows:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (2)$$

where $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ denotes the observation noise. The observation noise is considered to have normal distribution which has zero mean and covariance matrix \mathbf{R}_k . The \mathbf{z}_k represents observation vector. The h represents a non-linear function which describes the relationship between the state information vector \mathbf{x}_k and observation vector \mathbf{z}_k . The EKF algorithm is based on two important phases: (i) 'State Predict' and (ii) 'State Update'.

3.1. State Predict

As the name implies, this phase deals with the prediction of a priori state information vector $\hat{\mathbf{x}}_{k|k-1}$ using information from the previous a posteriori state information vector $\hat{\mathbf{x}}_{k-1|k-1}$. The state predict phase can be described as follows:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F} \cdot \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \cdot \mathbf{u}_k, \quad (3)$$

where \mathbf{u}_k is system input, \mathbf{F} represents state transition matrix and \mathbf{B}_k denotes input matrix. The covariance matrix can be obtained as follows:

$$\mathbf{P}_{k|k-1} = \mathbf{F} \cdot \mathbf{P}_{k-1|k-1} \cdot \mathbf{F}^T + \mathbf{Q}, \quad (4)$$

where $\mathbf{P}_{k|k-1}$ and $\mathbf{P}_{k-1|k-1}$ are respectively related to current state information and previous covariance matrix. \mathbf{Q} denotes the covariance matrix of process noise.

3.2. State Update

This phase is also known as refinement phase or correction phase of EKF. A priori state information estimates are refined further during this phase by utilizing the observation vector. The innovation vector is represented with $\tilde{\mathbf{y}}_k$ and can be calculated as follows:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_k), \quad (5)$$

where $h(\hat{\mathbf{x}}_k)$ represents the expected measurements. The innovation vector $\tilde{\mathbf{y}}_k$ covariance matrix \mathbf{S}_k can be computed as follows:

$$\mathbf{S}_k = \mathbf{H}_k \cdot \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^T + \mathbf{R}_k, \quad (6)$$

where \mathbf{H}_k represents Jacobian matrix of expected measurements $h(\hat{\mathbf{x}}_k)$. Now, a posteriori state vector $\hat{\mathbf{x}}_{k|k}$ is calculated as follows:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \cdot \tilde{\mathbf{y}}_k, \quad (7)$$

where \mathbf{K}_k denotes the optimal Kalman gain and is computed as follows:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^T \cdot \mathbf{S}_k^{-1}. \quad (8)$$

Now a posteriori state covariance matrix can be denoted with $\mathbf{P}_{k|k}$ and is computed as follows:

$$\mathbf{P}_{k|k} = (\mathbf{I}_n - \mathbf{K}_k \cdot \mathbf{H}_k) \cdot \mathbf{P}_{k|k-1}. \quad (9)$$

4. Positioning Through Extended Kalman Filter

A set of range measurements is required for localization of a mobile node in WSNs. This process involves calculation of time difference of arrival between the ultrasound and RF signals transmitted synchronously by a set of anchor nodes/beacons to a mobile node. We have limited our experiments to two dimensional (2D) space. All anchor nodes/beacons are places at pre-defined fixed location. A generic i -th beacon can be represented as $\mathbf{x}_{Anch_i} = [x_{Anch_i}, y_{Anch_i}]^T$. For the purpose of generalization, let L be the total number of beacons. Thus, the observation vector \mathbf{z}_k for 2D space can be expressed as follows:

$$\mathbf{z}_k = [\tilde{d}_{Anch_{1,k}} \quad \tilde{d}_{Anch_{2,k}} \quad \dots \quad \tilde{d}_{Anch_{L,k}}]^T, \tag{10}$$

where $\tilde{d}_{Anch_{i,k}}$ denotes the estimated distance of i -th beacon from the mobile node. The expected measurement vector can be expressed as Euclidean distances of all beacons from the mobile node as follows:

$$h(\hat{\mathbf{x}}_k) = \begin{bmatrix} dist(\hat{\mathbf{x}}_{Mob,k}, \mathbf{x}_{Anch_1}) \\ dist(\hat{\mathbf{x}}_{Mob,k}, \mathbf{x}_{Anch_2}) \\ \vdots \\ dist(\hat{\mathbf{x}}_{Mob,k}, \mathbf{x}_{Anch_L}) \end{bmatrix}, \tag{11}$$

where $\hat{\mathbf{x}}_{Mob,k} = [\hat{x}_{Mob,k}, \hat{y}_{Mob,k}]^T$ denotes the coordinates/position of mobile node while Euclidean distance is denoted with $dist(\cdot)$. For i -th beacon, the Euclidean distance can be computed as follows:

$$dist(\hat{\mathbf{x}}_{Mob,k}, \mathbf{x}_{Anch_i}) = \sqrt{(\hat{x}_{Mob,k} - x_{Anch_i})^2 + (\hat{y}_{Mob,k} - y_{Anch_i})^2} \tag{12}$$

Now the Jacobian matrix \mathbf{H}_k in (6) can be expressed as follows:

$$\mathbf{H}_k = \begin{bmatrix} \frac{\hat{x}_{Mob,k} - x_{Anch_1}}{dist(\hat{\mathbf{x}}_{Mob,k}, \mathbf{x}_{Anch_1})} & \frac{\hat{y}_{Mob,k} - y_{Anch_1}}{dist(\hat{\mathbf{x}}_{Mob,k}, \mathbf{x}_{Anch_1})} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\hat{x}_{Mob,k} - x_{Anch_L}}{dist(\hat{\mathbf{x}}_{Mob,k}, \mathbf{x}_{Anch_L})} & \frac{\hat{y}_{Mob,k} - y_{Anch_L}}{dist(\hat{\mathbf{x}}_{Mob,k}, \mathbf{x}_{Anch_L})} & 0 & 0 \end{bmatrix} \tag{13}$$

Finally, the covariance matrix \mathbf{R}_k in (6) can be expressed as follows:

$$\mathbf{R}_k = diag(\sigma_{d_{Anch_{1,k}}}^2 \quad \dots \quad \sigma_{d_{Anch_{L,k}}}^2), \tag{14}$$

where $\sigma_{d_{Anch_{i,k}}}^2$ denotes the initial variance in the distance measurement obtained for the i -th beacon. The experiments were performed using three different state models: ‘P Model’, ‘PV Model’ and ‘PVA Model’.

4.1. P-Model

The P model uses only position components in the algorithm. The state vector \mathbf{x}_k can be defined as follows:

$$\mathbf{x}_k = [x_k \quad y_k]^T.$$

Assuming the system with no input, we can represent the state transition matrix \mathbf{F} as follows:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{15}$$

Now we can express the covariance matrix for the process noise as follows:

$$\mathbf{Q} = \begin{bmatrix} \sigma_{v_x}^2 & 0 \\ 0 & \sigma_{v_y}^2 \end{bmatrix}, \tag{16}$$

where $\sigma_{v_x}^2$ and $\sigma_{v_y}^2$ respectively denotes noise variances along x and y directions. Finally, we can represent the covariance matrix \mathbf{P}_0 for the initial state information vector as follows:

$$\mathbf{P}_0 = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \tag{17}$$

where σ_x^2 and σ_y^2 represent variances of the state vector components.

4.2. PV-Model

The PV model uses both, position and velocity components in the algorithm. Thus, the state vector \mathbf{x}_k contains position and velocity components and can be defined as follows:

$$\mathbf{x}_k = [x_k \quad y_k \quad v_{x_k} \quad v_{y_k}]^T,$$

where v_{x_k} and v_{y_k} respectively denotes mobile node speed in the x and y directions. Assuming the system

with no input ($\mathbf{u}_k=0$), we can represent the state transition matrix \mathbf{F} as follows:

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_2 & \Delta t \cdot \mathbf{I}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix}, \tag{18}$$

where \mathbf{I}_2 represents identity matrix while \mathbf{O}_2 represent a zero matrix. The time interval between previous and current state estimate is represented by Δt . Now we can expressed the covariance matrix for the process noise as follows:

$$\mathbf{Q} = \mathbf{A} \cdot \begin{bmatrix} \sigma_{a_x}^2 & 0 \\ 0 & \sigma_{a_y}^2 \end{bmatrix} \cdot \mathbf{A}^T, \tag{19}$$

where $\sigma_{a_x}^2$ and $\sigma_{a_y}^2$ respectively denote acceleration noise variances in the x and y directions. Now we can expressed the matrix \mathbf{A} as follows:

$$\mathbf{A} = \begin{bmatrix} \frac{\Delta t^2}{2} \cdot \mathbf{I}_2 \\ \Delta t \cdot \mathbf{I}_2 \end{bmatrix}.$$

Finally, we can represent the covariance matrix \mathbf{P}_0 for the initial state information vector as follows:

$$\mathbf{P}_0 = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v_y}^2 \end{bmatrix}, \tag{20}$$

where $\sigma_x^2, \sigma_y^2, \sigma_{v_x}^2$ and $\sigma_{v_y}^2$ represent variances of the state vector components.

4.3. PVA-Model

The PVA model also uses acceleration bias values along with the position and velocity components in the algorithm. Thus, the state vector \mathbf{x}_k contains position, velocity and acceleration components and can be defined as follows:

$$\mathbf{x}_k = [x_k \quad y_k \quad v_{x_k} \quad v_{y_k} \quad \delta a_{x_k} \quad \delta a_{y_k}]^T,$$

where δa_{x_k} and δa_{y_k} respectively denotes measured acceleration values bias from true values in the x and y directions. The acceleration value is provided as an input ($\mathbf{u}_k = [a_{x_k} \quad a_{y_k}]^T$) to the system. We can express the input matrix \mathbf{B}_k as follows:

$$\mathbf{B}_k = \begin{bmatrix} \frac{\Delta t^2}{2} \cdot \mathbf{I}_2 \\ \Delta t \cdot \mathbf{I}_2 \\ \mathbf{O}_2 \end{bmatrix}.$$

We can represent state transition matrix for PVA model as follows:

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_2 & \Delta t \cdot \mathbf{I}_2 & \frac{\Delta t^2}{2} \cdot \mathbf{I}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 & \Delta t \cdot \mathbf{I}_2 \\ \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix}, \tag{21}$$

Now we can represent matrix \mathbf{A} in the equation of covariance matrix for process noise as follows:

$$\mathbf{A} = \begin{bmatrix} \frac{\Delta t^2}{2} \cdot \mathbf{I}_2 \\ \Delta t \cdot \mathbf{I}_2 \\ \mathbf{I}_2 \end{bmatrix}.$$

Finally, we can represent the covariance matrix \mathbf{P}_0 for the initial state information vector as follows:

$$\mathbf{P}_0 = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{v_x}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_y}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\delta a_x}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\delta a_y}^2 \end{bmatrix}, \tag{22}$$

where $\sigma_x^2, \sigma_y^2, \sigma_{v_x}^2, \sigma_{v_y}^2, \sigma_{\delta a_x}^2$ and $\sigma_{\delta a_y}^2$ represent variances of the state vector components.

5. Experimental Results

We evaluated the performance of EKF for P model, PV model and PVA model. The experiment was performed in indoor environment and beacons were placed in rectangular manner as depicted in Fig. 4(a). All of these measurements were manually taken and are subject to human error. Thus, the simulation is based on the 2-dimensional space where beacons/sensors are deployed on the floor. The distance measurements returned by the sensors usually fluctuate as the ultrasound usually introduces noise. The cricket listener attached to mobile node is moved on the predefined path as shown in Fig. 4(a). The initial location was calculated by trilateration and then trajectory of the mobile node is predicted using EKF algorithm. Fig. 4(b) depicts the percentage error distribution in the position estimation for P model, PV model and PVA model. The error distribution is Gaussian and almost similar for all the three models. However, still it can be observed in Fig. 4(b) that P model has higher percentage for small magnitude errors compared to PV model. Further, PVA model has higher percentage for large magnitude errors compared to other models. Likewise, the PV model has higher percentage errors of low magnitudes.

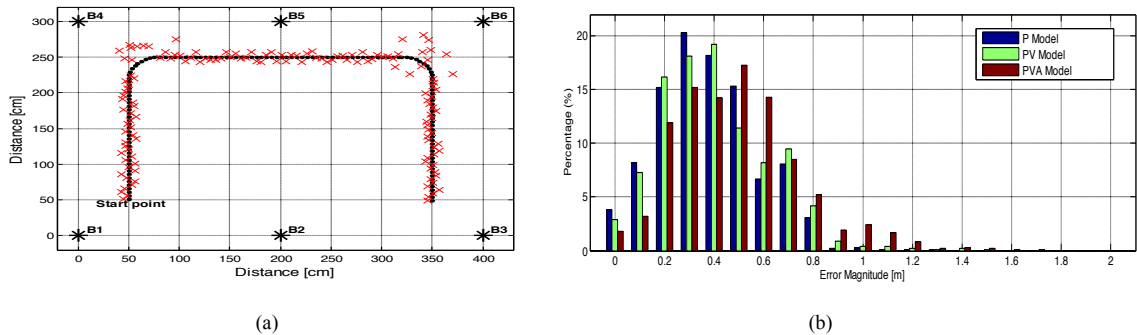


Fig. 4. (a) Simulation window, (b) Percentage error distribution.

We have evaluated the positioning and tracking performance of EKF under varying speed scenarios of mobile node on predefined track and can be observed in Table I. Table I represents the performance in Root Mean Square Error (RMSE). It can be observed that the performance of P model is comparatively better than PV and PVA models. Also PV model has slightly better performance compared to PVA model due to low accuracy of acceleration measurements. The effect of the speed of the mobile node is quite small and can also be observed in the Table I. The performance is better for low speed that introduces small errors compared to high speed scenarios. The results can be better observed in Fig. 5(a). The localization performance of P model, PV model and PVA model for different number of beacons/anchor nodes is presented in Table II. It can be observed that positioning performance is better when more beacons are used. Thus, the localization performance increases with the increase in the number of beacons and their specific location. Further, the performance of P model and PV model is better compared to PVA model. Graphically, it can be observed from Fig. 5(b) that the performance of P model, PV model and PVA model is slightly different under varying number of beacons.

TABLE I. COMPARISON OF P MODEL, PV MODEL AND PVA MODEL.

Model	RMSE [m]		
	20 cm/s	40 cm/s	60 cm/s
P	0.358	0.362	0.373
PV	0.431	0.447	0.462
PVA	0.493	0.534	0.561

TABLE II. PERFORMANCE COMPARISON UNDER VARYING NUMBER OF BEACONS.

No. of beacons	Beacons	RMSE [m]		
		P	PV	PVA
3	B1,B3,B4	0.42	0.483	0.572
4	B1,B3,B4,B6	0.361	0.436	0.51
6	B1,B2,B3, B4,B5,B6	0.358	0.431	0.493

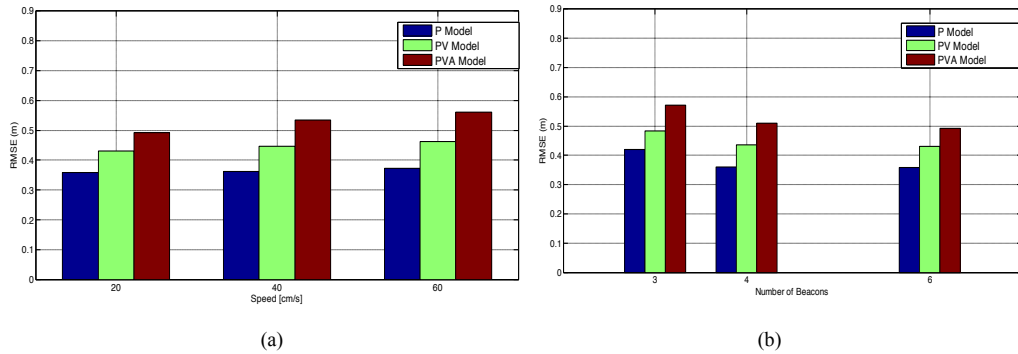


Fig. 5. (a) Performance comparison at varying speed, (b) Performance comparison with varying beacons.

6. Conclusions

This paper has presented in details the EKF state equations for two dimensional system and evaluated its localization performance for P model, PV model and PVA model under varying number of beacons and different mobile node speeds. The EKF is quite suitable for localization in wireless sensors network even if the distance measurements are affected by noise. The simulation concluded that P model performance is better compared to PV and PVA models. However, the performance relies on the accuracy of velocity and acceleration data. The P model is quite suitable when the mobile trajectory has sharp turns as it is less bound by velocity and acceleration and treat them as random noise. However, PV and PVA models are more affected by discontinuous velocity and acceleration measurements, respectively. Thus, the performance of PVA model is low as the acceleration measurements are affected by noise. Further, it can be observed that the localization performance also depends on the mobile node speed. The performance is better at low speed due to less induced noise/errors. Also the location performance depends on the number of beacons/anchor nodes. The localization performance is better if more beacons are used and well distributed in the experimental space.

References

- [1] X. Bin, C. Lin, X. Qingjun and L. Minglu, "Reliable Anchor-Based Sensor Localization in Irregular Areas", in IEEE Transactions on Mobile Computing, Vol. 9, Issue 1, Pages: 60 – 72, Jan. 2010.
- [2] C. Xunaue, S. Zhiguan, and L. Jianjun, "Distributed localization for anchor-free sensor networks", in Systems Engineering and Electronics Journal, Vol. 19, Issue. 3, Pages: 405 - 418, June 2008.
- [3] N. B Priyantha, H. Balakrishnan, E. Demaine, S. Teller, "Anchor-Free Distributed Localization in Sensor Networks", Tech. Rep. 892, MIT. Lab. for Computer Science, April 2003.
- [4] S. Thrun, "Robotic mapping: A survey", in Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann, 2002.
- [5] P. Chalimbaud, F. Marmoiton, and F. Berry, "Towards an embedded visuo-inertial smart sensor", International Journal in Robotic Research, 2007.
- [6] K. R. Anne, K. Kyamakya, F. Erbas, C. Takenga, and J. C. Chedjou, "GSM RSSI- based positioning using extended Kalman filter for training ANN," in IEEE Vehicular Technology Conference (VTC), 2004.
- [7] J. Albuquerque, J.L. Lair, B. Govin, G. Muller and P. Riant, "Autonomous satellite navigation using optico inertial instruments", Automatic control in space, pp. 183-188, 1985.
- [8] R.R. Pinho, J.M.R.S. Tavares, "Comparison between Kalman and Unscented Kalman Filters in Tracking Applications of Computational Vision", in VipIMAGE 2009.
- [9] S. J. Julier and J. K. Uhlmann, "Reduced Sigma Point Filters for the Propagation of Means and Covariances Through nonlinear Transformations", in Proc. American Control Conference Alaska, pp.887-892, USA, 2002.
- [10] R. Merwe, A. Doucety, N. Freitas, E. Wan, "The Unscented Kalman Filter Advances" in Neural Information Processing Systems 2000, Vancouver, Canada.
- [11] G. F. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina, Chapel Hill, NC, USA, Tech. Rep., 1995.