Navigation of Robotic System using Cricket Motes

Yogendra J. Patil[a], Nicholas A[b]. Baine, Kuldip S. Rattan[c]

Wright State University, Department of Electrical Engineering, Dayton, OH 45435

[a]w001yjp@wright.edu, [b]nicholas.baine@wright.edu, [c]kuldip.rattan@wright.edu.

ABSTRACT

This paper presents a novel algorithm for self-mapping of the cricket motes that can be used for indoor navigation of autonomous robotic systems. The cricket system is a wireless sensor network that can provide indoor localization service to its user via acoustic ranging techniques. The behavior of the ultrasonic transducer on the cricket mote is studied and the regions where satisfactorily distance measurements can be obtained are recorded. Placing the motes in these regions results fine-grain mapping of the cricket motes. Trilateration is used to obtain a rigid coordinate system, but is insufficient if the network is to be used for navigation. A modified SLAM algorithm is applied to overcome the shortcomings of trilateration. Finally, the self-mapped cricket motes can be used for navigation of autonomous robotic systems in an indoor location.

Keywords: Cricket motes, Ultrasonic (US) pulse, Radio Frequency (RF) signal, Time Difference of Arrival (TDoA), Trilateration, Extended Kalman Filtering (EKF), Simultaneous Localization And Mapping (SLAM) algorithm, Kalman Filter.

1. INTRODUCTION

To navigate through an area and avoid known obstacles, an autonomous robot requires the knowledge of its location. In outdoor environments, the Global Positioning System (GPS) can be used to provide precise mapping of a location. Though GPS is a reliable source for global positioning, it has several limitations that cannot be overlooked. Limitations such as high power consumption of the GPS unit, degree of accuracy is dependent on number of visible satellites, non-availability during national-security emergency and degradation of quality of signal due to atmospheric electronic interferences. The main limitation of GPS is that its signals are degraded in indoor environments and therefore, GPS cannot provide indoor navigation [1][2]. Cricket motes can provide indoor mapping and localization, which is difficult to achieve using GPS [3].

Cricket motes were developed at Computer Science and Artificial Lab (CSAIL), MIT as a low cost, low power-operated location-aware system. These motes are similar to micaz motes, which can be programmed using TinyOS, NesC platform. Cricket motes operate in two modes – beacon mode and listener mode. A mote in beacon mode periodically transmits an ultrasonic (US) pulse and a radio frequency (RF) signal, while a cricket mote in listener mode receives the US pulse and RF signal transmitted by the beacon. In short, cricket motes act as transmitters in beacon mode and as receivers in listener mode. Beacons are usually mounted on the ceiling with user-configured coordinates, while a listener is attached to the user device. The location of the listener with respect to a given set of

beacons is calculated on the user device. Cricket mote use Time Difference of Arrival (TDoA) measurements to obtain ranging between a beacon and a listener [3].

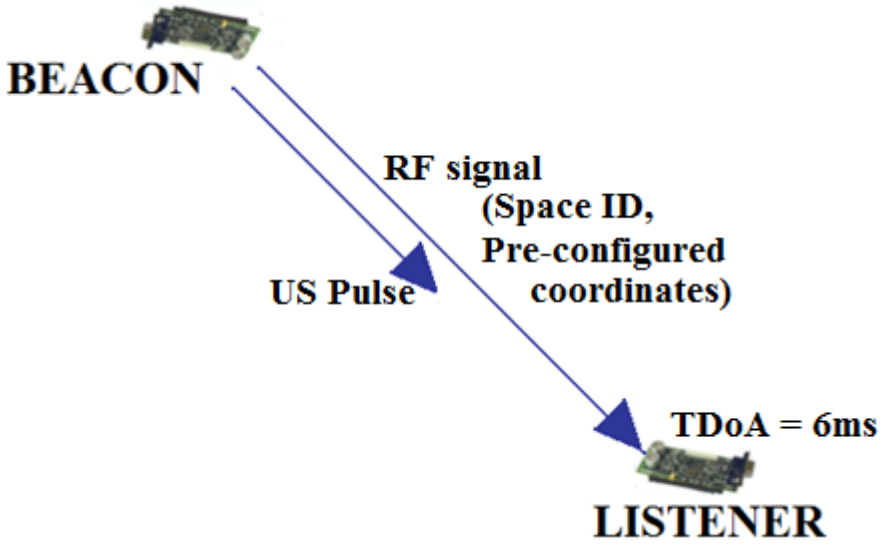

Figure1. Time Difference of Arrival (TDoA) measurement.

Figure 1 shows the general idea for calculating the distance between a beacon and a listener using the principle of the TDoA technique. In TDoA technique, beacon transmits the RF signal (carrying beacon location information) and US pulse simultaneously. RF signals travels at the speed of light ($3x10^8$ m/s) while the US signal travels at the speed of sound (344 m/s at 20℃). Therefore, the listener will receive the RF signal first and then the US pulse. Listener is programmed to record the RF signal arrival time, t_1 and the US pulse arrival time, t_2. Calculating the difference of t_1 and t_2, gives the TDoA measurement between the RF and the US signal. Cricket mote has an on-board temperature sensor, which calculates the ambient temperature [3]. Using the knowledge of the recorded ambient temperature, cricket mote calculates the corresponding speed of sound V_{US}. Distance between the beacon and the listener d_{BL} is obtained at the listener end using equation (1),

$$d_{BL} = V_{US} \cdot (t_2 - t_1) = V_{US} \cdot \Delta T \quad (1)$$

For example, if ΔT = 0.006s and speed of sound V_{US} = 34400 cm/s, then the value of d_{BL} is given by

$$d_{BL} = 34400 \text{ x } 0.006 = 206.4 \text{ cm}$$

Once the listener calculates three distances from three different beacons, it can calculate its position coordinates using a method of trilateration [2][4]. If the unknown point is X (x, y) with three distance measurements d_{AX}, d_{BX}, d_{CX} from known coordinates A (x_a, y_a), B (x_b, y_b) and C (x_c, y_c), respectively, then the point X can be uniquely localized by solving equations (2), (3) and (4) simultaneously.

$$(x - x_a)^2 + (y - y_a)^2 = (d_{AX})^2 \quad (2)$$

$$(x - x_b)^2 + (y - y_b)^2 = (d_{BX})^2 \quad (3)$$

$$(x - x_c)^2 + (y - y_c)^2 = (d_{CX})^2 \quad (4)$$

Equations (2), (3) and (4) represent equation of three circles, with position of unknown point X as their common point of intersection. Trilateration is the most widely accepted technique for localization because of its reduced computational complexity as compared to other methods [5]. Though trilateration provides a rigid mapping, factors such as ambiguity of distance measurements and geometric layout of sensor motes play a crucial role in the accuracy of the final mapping. The latter mentioned factor is usually referred as Geometric Dilution of Precision (GDoP). This is similar to the GDoP factor observed for GPS satellites [1]. The errors in the distance calculation can never be eliminated because no ranging instruments are perfect and have inaccuracies and uncertainty with inherent noise. Due to this problem, the mapping of the points is in-accurate and the final graph formed due to this mapping highly differs from the actual graph representation.

The solution to this problem is obtained by simply applying the Simultaneous Localization And Mapping (SLAM) algorithm [6]. SLAM algorithms are widely used in the field of autonomous robot navigation, especially in scenarios where a robot encounters an unknown environment with its own location unknown. Using SLAM, the robot moves through an unknown environment taking observations of the unknown landmarks and tries to relate each observation to its respective current state (position and orientation). The error in the locations of each landmark due to noisy observations is reduced by applying the Extended Kalman Filtering (EKF) technique [7]. In this paper, a modified SLAM algorithm is used to map the cricket motes. Kalman Filter (KF) technique is applied, in situations where measurements (observations) observed over time have inaccuracies [8].

2. Kalman Filter

The Kalman Filter (KF) technique is used to estimate the true value of the states $\boldsymbol{x} \in \mathfrak{R}^n$ defined by the linear stochastic difference equation,

$$\boldsymbol{x}_k = \boldsymbol{A}\,\boldsymbol{x}_{k-1} + \boldsymbol{B}\,\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1} \tag{5}$$

by using a measurement $\boldsymbol{z} \in \mathfrak{R}^m$ given by,

$$\boldsymbol{z}_k = \boldsymbol{H}_k\,\boldsymbol{x}_k + \boldsymbol{v}_k \tag{6}$$

where, $\boldsymbol{x}_k$ is the state vector of the system at k^{th} observation (measurement or iteration),

$\boldsymbol{x}_{k-1}$ is the state vector of the system at $k-1^{th}$ observation (measurement or iteration),

$\boldsymbol{A}$ is the state transition matrix describing the state transition from k^{th} to $k-1^{th}$ observation,

$\boldsymbol{B}$ is the matrix relating control inputs to the state of the system,

$\boldsymbol{u}_{k-1}$ is the control input applied at $k-1^{th}$ observation,

$\boldsymbol{z}_k$ is the k^{th} observation (measurement or iteration),

$\boldsymbol{H}$ is the matrix relating the state ($\boldsymbol{x}_k$) to the measurement ($\boldsymbol{z}_k$),

$\boldsymbol{w}_{k-1}$, $\boldsymbol{v}_k$ represents the process noise and the measurement noise for the given observation, respectively,

with normal probability distribution function given by

$\boldsymbol{p}(\boldsymbol{w}) \sim N(0, \boldsymbol{Q})$, where $\boldsymbol{Q}$ is the process noise covariance and

$\boldsymbol{p}(\boldsymbol{v}) \sim N(0, \boldsymbol{R})$, where $\boldsymbol{R}$ is the measurement noise covariance.

The following nomenclature is used in obtaining the KF technique,

$\hat{\boldsymbol{x}}^{-}_{k}$ represents *a priori* (minus sign as superscript) state estimates (cap) for the observation k (at subscript)

$\hat{\boldsymbol{x}}_{k}$ represents a posteriori (no sign at superscript) state estimates (cap) for the observation k (at subscript)

The a priori and a posteriori error estimates are given by,

$$\mathbf{e}^{-}_{k} \approx \boldsymbol{x}_{k} - \hat{\boldsymbol{x}}^{-}_{k}, \text{ and} \tag{7}$$

$$\mathbf{e}_{k} \approx \boldsymbol{x}_{k} - \hat{\boldsymbol{x}}_{k} \tag{8}$$

The corresponding a priori and a posteriori error covariance estimate are given by,

$$\boldsymbol{P}^{-}_{k} = \mathbf{E}[\mathbf{e}^{-}_{k}\, \mathbf{e}^{-\mathrm{T}}_{k}] \text{ and} \tag{9}$$

$$\boldsymbol{P}_{k} = \mathbf{E}[\mathbf{e}_{k}\, \mathbf{e}_{k}^{\mathrm{T}}] \tag{10}$$

The objective of the KF is to find a posteriori estimate $\hat{\boldsymbol{x}}_{k}$ using the knowledge of the $\hat{\boldsymbol{x}}^{-}_{k}$, the given observation $\boldsymbol{z}_{k}$ and the estimated observation value $\boldsymbol{H}\,\hat{\boldsymbol{x}}^{-}_{k}$, which is defined as,

$$\hat{\boldsymbol{x}}_{k} = \hat{\boldsymbol{x}}^{-}_{k} + \boldsymbol{K}_{k}\,(\boldsymbol{z}_{k} - \boldsymbol{H}\,\hat{\boldsymbol{x}}^{-}_{k}) \tag{11}$$

The term $(\boldsymbol{z}_{k} - \boldsymbol{H}\,\hat{\boldsymbol{x}}^{-}_{k})$ is usually defined as the residual or the innovative factor [8]. The Kalman filter gain or the blending factor $\boldsymbol{K}$ is given by,

$$\boldsymbol{K}_{k} = \boldsymbol{P}^{-}_{k}\,\boldsymbol{H}^{\mathrm{T}} / (\boldsymbol{H}\boldsymbol{P}^{-}_{k}\,\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}) \tag{12}$$

The idea behind finding the value of $\boldsymbol{K}$ is to assign the residual factor, given in equation (11), a value such that the resulting a posteriori state vector estimate $\hat{\boldsymbol{x}}_{k}$ has a lower error covariance than its a priori state estimate value $\hat{\boldsymbol{x}}^{-}_{k}$ [8].

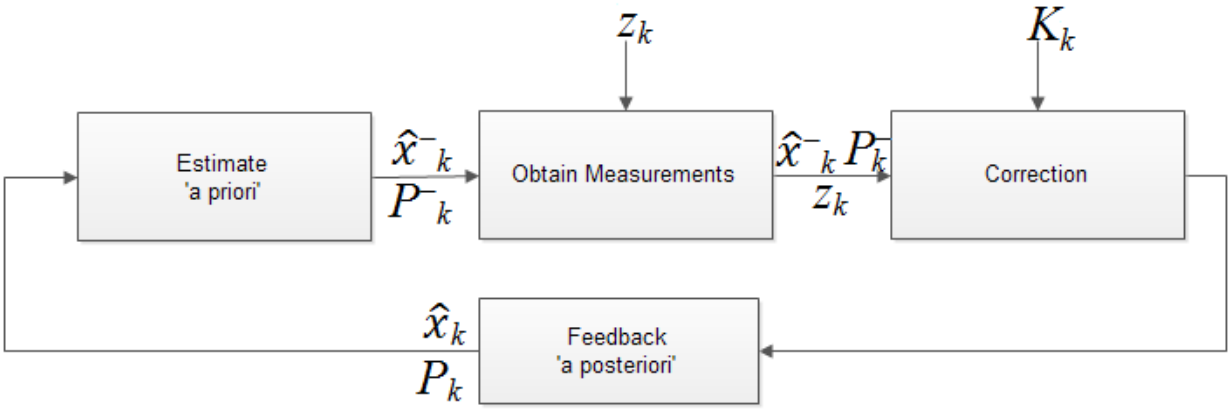

Figure 2. Flow Chart for Kalman Filter Process.

Figure 2 describes the flow chart for the Kalman filtering process for k^{th} observation to find the 'a posteriori' values for state and error covariance using the knowledge of the 'a priori' values of the state and error covariance, measurements and the (calculated) Kalman gain. The flow chart consists of following steps,

- Estimate 'a priori': Estimate the a priori values of the state $\hat{\boldsymbol{x}}^{-}_{k}$ and the error covariance $\boldsymbol{P}^{-}_{k}$ given by the equations,

$$\hat{\boldsymbol{x}}^{-}_{k} = \boldsymbol{A}\,\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\,\boldsymbol{u}_{k-1} \tag{13}$$

$$\boldsymbol{P}^{-}_{k} = \boldsymbol{A}\,\boldsymbol{P}^{-}_{k-1}\,\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{Q} \tag{14}$$

- Obtain Measurements: Obtain measurement for the k^{th} iteration
- Correction: Calculate the Kalman filter gain and (update) the 'a posteriori' state estimate using the following equations,

$$\boldsymbol{K}_{k} = \boldsymbol{P}^{-}_{k}\,\boldsymbol{H}^{\mathrm{T}}\,(\boldsymbol{H}\boldsymbol{P}^{-}_{k}\,\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \tag{15}$$

$$\hat{\boldsymbol{x}}_{k} = \hat{\boldsymbol{x}}^{-}_{k} + \boldsymbol{K}_{k}\,(\boldsymbol{z}_{k} - \boldsymbol{H}\,\hat{\boldsymbol{x}}^{-}_{k}) \tag{16}$$

$$\boldsymbol{P}_{k} = (\boldsymbol{I} - \boldsymbol{K}_{k}\,\boldsymbol{H})\,\boldsymbol{P}^{-}_{k} \tag{17}$$

- Feedback 'a posteriori': Feedback the estimated state and the error covariance values to the estimation stage of the Kalman filter.

The Kalman filter consists of two steps,

- Estimation stage: Estimate the values of the (a priori) state and the error covariance matrix and forward it to the measurement stage.
- Measurement stage: Obtain the (noisy) measurement value, and update the value of the states using the knowledge of the a priori state, error covariance and measurements obtained.

In this paper the Extended Kalman Filter (EKF) technique is used which applies the Kalman filter technique to the non-linear systems to linearize the process about the current mean and covariance [6][7].

3. Algorithm Formulation

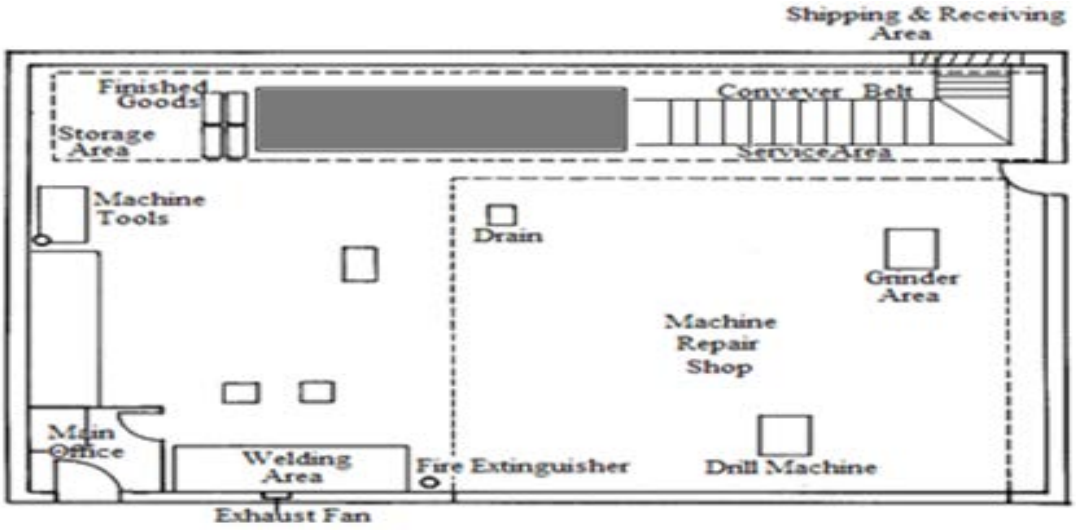

Figure 6. Floor Plan of a Manufaturing Factory with shaded region to be mapped.

Figure 6 shows a floor plan for a manufacturing factory where an autonomous robot (say an autonomous fork lift) needs to transport the finished goods from the storage area to the conveyer belt in the service area. The autonomous robot needs its location information at every instance to navigate to the destination. Figure 6 shows a shaded region (9m x 5m) between finished goods and conveyer belt, which is to be mapped using the cricket motes.

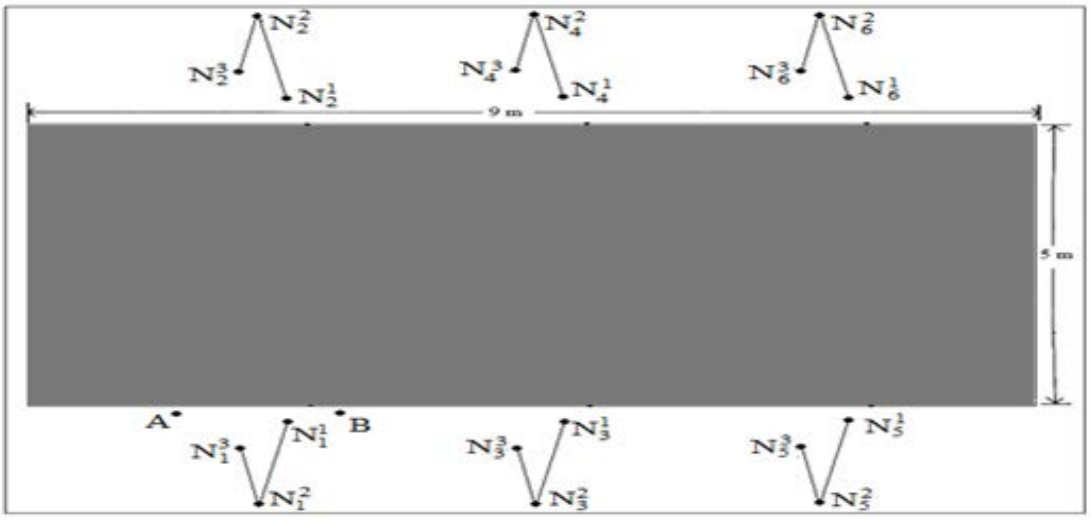

Figure 7. Top View of the field to be mapped.

Figure 7 shows the top view of the shaded region to be mapped and the layout of the cricket motes around the region. The cricket motes are mounted on a stand 1 meter above ground level and are maintained in the same plane. In the following discussion the network and the motes are referred using the following nomenclature,

$\boldsymbol{N}^{j}_{i}$: represents the j^{th} mote in i^{th} network, where i = 1, 2, 3, 4, 5, 6 and j = 1, 2, 3 (in this project) with coordinates, (x^{j}_{i}, y^{j}_{i}). For example, $N^{1}{}_{1}$ represents the mote of network 1 with coordinates $(x^{1}{}_{1}, y^{1}{}_{1})$.

Each network, N_i consists of a group of 3 motes $(N^{1}_{i}, N^{2}_{i}, N^{3}_{i})$. In every network the three motes first map themselves and then assist in mapping of the other motes. Each mote is programmed to be mapped only by the motes from a given network, using the method of trilateration. For example, motes $N^{1}{}_{2}$, $N^{2}{}_{2}$, $N^{3}{}_{2}$ in network N_2 are programmed to be mapped by motes $N^{1}{}_{1}$, $N^{2}{}_{1}$, $N^{3}{}_{1}$ of network N_1, using the method of trilateration. Motes $N^{1}{}_{3}$, $N^{2}{}_{3}$, $N^{3}{}_{3}$ in network N_3 are programmed to be mapped by motes $N^{1}{}_{2}$, $N^{2}{}_{2}$, $N^{3}{}_{2}$ of network N_2, using the method of trilateration. In other words, mapping of the motes follows a zigzag path from network N_1 up to network N_6. Distance separation between each network is adjusted so that mote $N_{3}{}^{2}$ and $N_{5}{}^{2}$ are able to calculate its distance from motes $N_{2}{}^{2}$ and $N_{4}{}^{2}$ respectively (using the experimental field data).

It is advisable to keep network 1 and 2 at a distance of 1 meter from the beginning point, to avoid any US multi-path interference in case there is a wall nearby and also to increase the overall range capacity of the system [3]. For a given network, the inter mote distances are fixed and each mote knows its distance and orientation with respect to other motes, within the network. For example, consider a network N_i with motes N^{1}_{i}, N^{2}_{i}, N^{3}_{i}. Mote N^{2}_{i} knows its distance and orientation with respect to motes N^{1}_{i} and N^{3}_{i}. Motes A and B are the anchor motes, where A acts as the origin of the coordinate system and B as a point on Y-axis.

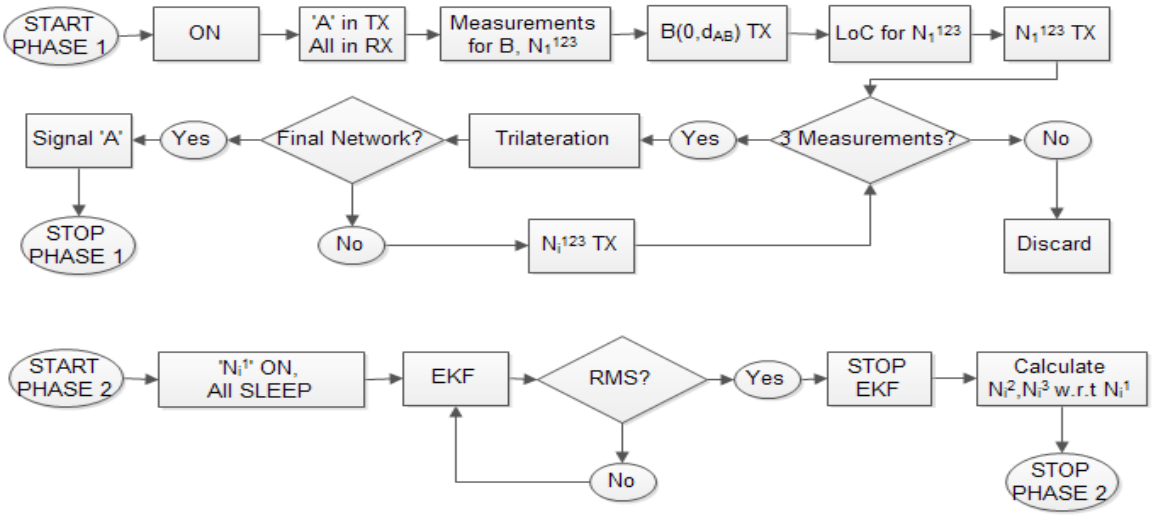

Figure 8. Flow Chart for the mapping of the given location.

In phase 1, the trilateration method is implemented to get rough estimates of the coordinates (states) of each mote-

- ON: Switch ON all motes.
- A in TX All in RX: The anchor mote 'A' in beacon mode will transmit (TX) its US and RF signal while all other motes in listener mode will receive (RX) these signals.
- Measurements for B, N_1^{123}: According to the experimental data only motes B, N_1^{123} will receive the US and RF pulse transmitted by mote A and will calculate their distance from A using the TDoA technique.
- B(0, d_{AB})TX: Mote B, the second anchor node, will configure its coordinate as (0, d_{AB}) i.e. The Y-axis of the coordinate system passes through mote B. d_{AB} is the distance between A and B, calculated using the TDoA technique. B transmits its US and RF signals.
- LoC for N_1^{123}: The three motes in network N_1 will calculate their coordinates using 'Law-of-Cosines'.
- N_1^{123} TX: The three motes in network N_1 will now transmit their US and RF signal.
- 3 Measurements: If three measurements (three coordinates and distances) are obtained from the required network, then calculate the coordinates using the method of trilateration.
- Final Network: If all the given networks are mapped, then signal anchor mote A to start phase 2 or else continue with the mapping.

In phase 2, the EKF technique is implemented to achieve high accuracy

- N_i^1 ON, All SLEEP: Mote N_i^1 of each network remains in ON state, while rest of the motes goes to sleep mode.
- EKF: The EKF technique is implemented using state and distance information between a pair of motes.
- RMS: Check if the desired root mean square (RMS) value of the error in mapping is achieved
- Calculate N_i^2, N_i^3 w.r.t N_i^1: Calculate the coordinates for motes N_i^2, N_i^3 with respect to the calculated values of N_i^1 and stop phase 2

In this paper, the EKF technique is applied because the relation between the states (coordinates) and the measurements (given by Euclidean distance formula) is non-linear [6].

In the following discussion, the variable n is defined as,

n = total number of networks x 2 (for 2D mapping)

= total number of networks x 3 (for 3D mapping)

In this paper 2D mapping is performed; therefore, n = 12. Equations (13) to (17) are given by,

$$\hat{\boldsymbol{x}}^{-}_{k} = \boldsymbol{A}\,\hat{\boldsymbol{x}}^{-}_{k-1} + \boldsymbol{B}\,\boldsymbol{u}_{k-1} \tag{18}$$

where, $\hat{\boldsymbol{x}}^{-}_{k-1\,(\mathrm{n\,x\,1})} = [\hat{x}^1_1\ \hat{y}^1_1\ \hat{x}^1_2\ \hat{y}^1_2\ \hat{x}^1_3\ \hat{y}^1_3\ \hat{x}^1_4\ \hat{y}^1_4\ \hat{x}^1_5\ \hat{y}^1_5\ \hat{x}^1_6\ \hat{y}^1_6]^{\mathrm{T}}_{(1\,\mathrm{x\,n})}$

x^j_i, y^j_i : represents the x and y coordinate for the j^{th} mote in i^{th} network. For $k = 1$, the values of these coordinates are obtained using the trilateration method. For $k > 1$, the values of the coordinates are obtained from the measurement phase.

and, $\boldsymbol{B}$, $\boldsymbol{u}_{k-1} = \mathbf{O}_{(\mathrm{n\,x\,1})}$, as no robot is used and the motes are stationary

$$\boldsymbol{A}_{(\mathrm{n\,x\,n})} = \mathbf{I}_{(\mathrm{n\,x\,n})} \tag{19}$$

$$= \begin{bmatrix} 1 & 0 \cdots & 0 \\ 0 & & \cdot \\ \cdot & \ddots & \cdot \\ \cdot & & 0 \\ 0 & \cdots & 0\ 1 \end{bmatrix} \tag{20}$$

From equations (18), (19) and (20),

$$\hat{\boldsymbol{x}}^{-}_{k} = \mathbf{I}\,\hat{x}^{-}_{k-1} + \mathbf{O} = \hat{\boldsymbol{x}}^{-}_{k-1} \tag{21}$$

$$\hat{\boldsymbol{x}}^{-}_{k\,(\mathrm{n\,x\,1})} = \hat{\boldsymbol{x}}^{-}_{k-1\,(\mathrm{n\,x\,1})} = [\hat{x}^1_1\ \hat{y}^1_1\ \hat{x}^1_2\ \hat{y}^1_2\ \hat{x}^1_3\ \hat{y}^1_3\ \hat{x}^1_4\ \hat{y}^1_4\ \hat{x}^1_5\ \hat{y}^1_5\ \hat{x}^1_6\ \hat{y}^1_6]^{\mathrm{T}}_{(1\,\mathrm{x\,n})} \tag{22}$$

$$\boldsymbol{P}^{-}_{k\,(\mathrm{n\,x\,n})} = \boldsymbol{A}_{(\mathrm{n\,x\,n})}\,\boldsymbol{P}^{-}_{k-1(\mathrm{n\,x\,n})}\,\boldsymbol{A}^{\mathrm{T}}_{(\mathrm{n\,x\,n})} + \boldsymbol{Q}_{(\mathrm{n\,x\,n})} \tag{23}$$

$\boldsymbol{Q} = \mathbf{O}_{(\mathrm{n\,x\,n})}$, as no robot is used and the motes are stationary (24)

$$\boldsymbol{P}^{-}_{k} = \begin{bmatrix} 0 & & & & \cdots & & 0 \\ \cdot & 0 & & & \cdots & & 0 \\ \cdot & & 99 & 0 & \cdots & & 0 \\ & & & 0 & & & \cdot \\ & & & & 99 & & \vdots \\ \vdots & & & & & \ddots & 0 \\ 0 & & & & \cdots & 0 & 99 \end{bmatrix} \tag{25}$$

Equation (25) gives the representation of the $\boldsymbol{P}^{-}_{k}$ matrix for $k = 1$. The first two and the fourth diagonal element of $\boldsymbol{P}^{-}_{k}$ are assigned a zero value because the accuracy in mapping the x and y coordinate of 1st network and y coordinate of 2nd network is high. From, equations (23) and (24)

$$\boldsymbol{P}^{-}_{k\,(\mathrm{n\,x\,n})} = \mathbf{I}\,\boldsymbol{P}^{-}_{k-1}\,\boldsymbol{I}^{\mathrm{T}} + \mathbf{O} = \boldsymbol{P}^{-}_{k-1} \tag{26}$$

In this algorithm, for a given iteration k, the TDoA measurement is taken between two motes N^1_p and N^1_q

$$\boldsymbol{K}_k = \boldsymbol{P}^{-}_{k}\,\boldsymbol{H}^{\mathrm{T}}\,(\boldsymbol{H}\boldsymbol{P}^{-}_{k}\,\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \tag{27}$$

The $\boldsymbol{H}$ matrix in equation (26) is given by,

$$\boldsymbol{H}_{p,q} = \left[\frac{\partial z_{p,q}}{\partial \hat{x}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{y}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{x}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{y}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{x}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{y}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{x}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{y}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{x}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{y}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{x}_1^1} \; \frac{\partial z_{p,q}}{\partial \hat{y}_1^1} \right] \tag{28}$$

where, $z_{p,q} = \sqrt{\left(\hat{x}_q^1 - \hat{x}_p^1\right)^2 + \left(\hat{y}_q^1 - \hat{y}_p^1\right)^2}$. (29)

If the TDoA measurement is taken between motes $N^1{}_1$ and $N^1{}_2$, then p = 1 and q = 2,

$$\boldsymbol{H}_{1,2} = \left[\frac{\partial z_{1,2}}{\partial \hat{x}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{y}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{x}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{y}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{x}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{y}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{x}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{y}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{x}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{y}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{x}_1^1} \; \frac{\partial z_{1,2}}{\partial \hat{y}_1^1} \right] \tag{30}$$

where, $$\frac{\partial z_{1,2}}{\partial \hat{x}_1^1} = \frac{\partial \sqrt{\left(\hat{x}_2^1 - \hat{x}_1^1\right)^2 + \left(\hat{y}_2^1 - \hat{y}_1^1\right)^2}}{\partial \hat{x}_1^1} = \frac{-\left(\hat{x}_2^1 - \hat{x}_1^1\right)}{\sqrt{\left(\hat{x}_q^1 - \hat{x}_p^1\right)^2 + \left(\hat{y}_q^1 - \hat{y}_p^1\right)^2}}, \tag{31}$$

$$\frac{\partial z_{1,2}}{\partial \hat{y}_1^1} = \frac{\partial \sqrt{\left(\hat{x}_2^1 - \hat{x}_1^1\right)^2 + \left(\hat{y}_2^1 - \hat{y}_1^1\right)^2}}{\partial \hat{y}_1^1} = \frac{-\left(\hat{y}_2^1 - \hat{y}_1^1\right)}{\sqrt{\left(\hat{x}_q^1 - \hat{x}_p^1\right)^2 + \left(\hat{y}_q^1 - \hat{y}_p^1\right)^2}}, \tag{32}$$

$$\frac{\partial z_{1,2}}{\partial \hat{x}_2^1} = \frac{\partial \sqrt{\left(\hat{x}_2^1 - \hat{x}_1^1\right)^2 + \left(\hat{y}_2^1 - \hat{y}_1^1\right)^2}}{\partial \hat{x}_2^1} = \frac{+\left(\hat{x}_2^1 - \hat{x}_1^1\right)}{\sqrt{\left(\hat{x}_q^1 - \hat{x}_p^1\right)^2 + \left(\hat{y}_q^1 - \hat{y}_p^1\right)^2}}, \tag{33}$$

$$\frac{\partial z_{1,2}}{\partial \hat{y}_2^1} = \frac{\partial \sqrt{\left(\hat{x}_2^1 - \hat{x}_1^1\right)^2 + \left(\hat{y}_2^1 - \hat{y}_1^1\right)^2}}{\partial \hat{y}_2^1} = \frac{+\left(\hat{y}_2^1 - \hat{y}_1^1\right)}{\sqrt{\left(\hat{x}_q^1 - \hat{x}_p^1\right)^2 + \left(\hat{y}_q^1 - \hat{y}_p^1\right)^2}}, \tag{34}$$

and $$\frac{\partial z_{1,2}}{\partial \hat{x}_3^1} = \frac{\partial z_{1,2}}{\partial \hat{y}_3^1} = \frac{\partial z_{1,2}}{\partial \hat{x}_4^1} = \frac{\partial z_{1,2}}{\partial \hat{y}_4^1} = \frac{\partial z_{1,2}}{\partial \hat{x}_5^1} = \frac{\partial z_{1,2}}{\partial \hat{y}_5^1} = \frac{\partial z_{1,2}}{\partial \hat{x}_6^1} = \frac{\partial z_{1,2}}{\partial \hat{y}_6^1} = 0 \tag{35}$$

Equations (30) to (35), give

$$\boldsymbol{H}_{1,2} = \left[\frac{-\left(\hat{x}_2^1 - \hat{x}_1^1\right)}{\sqrt{\left(\hat{x}_q^1 - \hat{x}_p^1\right)^2 + \left(\hat{y}_q^1 - \hat{y}_p^1\right)^2}} \; \frac{-\left(\hat{y}_2^1 - \hat{y}_1^1\right)}{\sqrt{\left(\hat{x}_q^1 - \hat{x}_p^1\right)^2 + \left(\hat{y}_q^1 - \hat{y}_p^1\right)^2}} \; \frac{+\left(\hat{x}_2^1 - \hat{x}_1^1\right)}{\sqrt{\left(\hat{x}_q^1 - \hat{x}_p^1\right)^2 + \left(\hat{y}_q^1 - \hat{y}_p^1\right)^2}} \; \frac{+\left(\hat{y}_2^1 - \hat{y}_1^1\right)}{\sqrt{\left(\hat{x}_q^1 - \hat{x}_p^1\right)^2 + \left(\hat{y}_q^1 - \hat{y}_p^1\right)^2}} \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \right] \tag{36}$$

The value of $\boldsymbol{R}$ obtained through field experiments, yielded $\sigma = 0.02/3 = 0.00666667$.

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}^{-}{}_k + \boldsymbol{K}_k \left(\boldsymbol{z}_k - \mathbf{z}_{\mathrm{p,q}}\right) \tag{37}$$

$$\boldsymbol{P}_k = \left(I - \boldsymbol{K}_k \boldsymbol{H}_{1,2}\right) \boldsymbol{P}^{-}{}_k \tag{38}$$

The a posteriori estimate $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{P}_k$ is again fedback to the step 1 of the estimation phase to repeat the process for state updation using the knowledge of another measurement (distance between two motes of two different networks).

4. Simulation Results

Using the data provided by the field experiments, phase 1 of the proposed algorithm is implemented to get the rough estimates of the position of each mote. Up to network 2 the error in mapping is very low, but the error increases slowly as we move towards the final network.

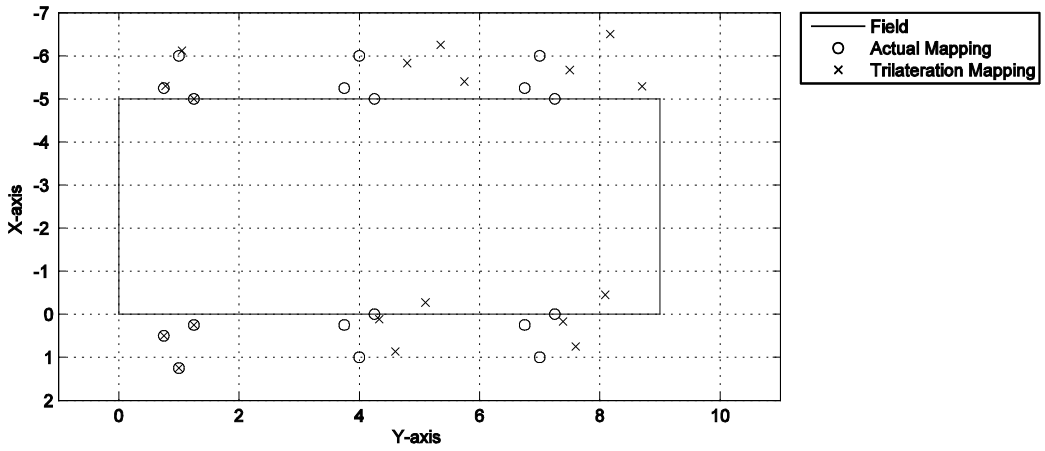

Figure 9. Errors in mapping of the motes using trilateration.

In Figure 9, the ‘rectangular box’ shows the outline of the field and the points marked with ‘circles’ are the actual position of the motes. Points marked with ‘cross’ represents the mapping of the motes due to method of trilateration, which has large amount of errors.

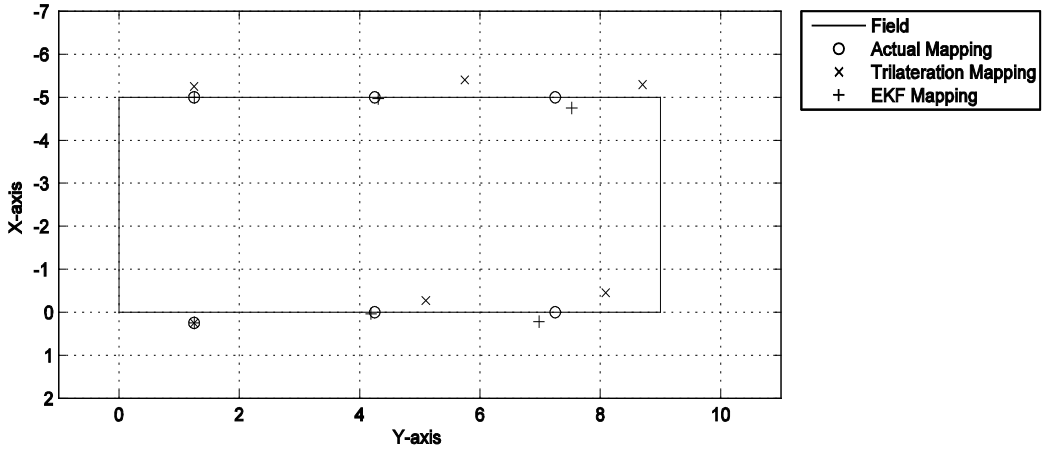

Figure 10. Corrections in Mapping using the EKF technique.

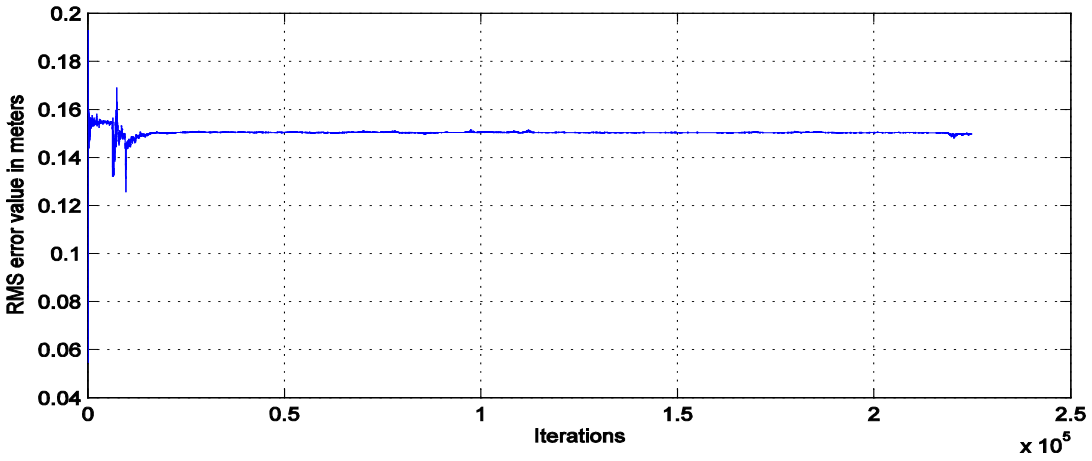

Figure 11. RMS error value for the state at each each iteration.

Figure 10 shows the scenario when phase 2 is implemented, considering only motes N_i^1 of each network. According to figure 10 motes N_1^1 and N_2^1 are mapped almost accurately, while there is some error in mapping of other motes. Figure 11 shows that the RMS error value starts to decrease rapidly at first few iterations and then starts to decrease at very slow pace. The RMS error value remains between 0.16 meters and 0.14 meters for much longer time. The experiment is repeated again, but for much larger iterations in order to achieve more accuracy.

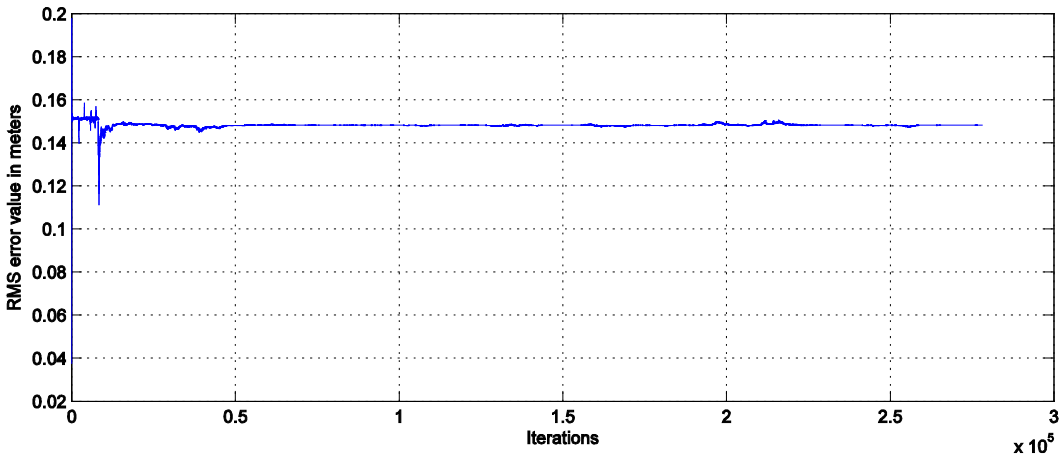

Figure 12. RMS error value at each each iteration.

Figure 12 shows that even after increasing the number of iterations, the RMS error value still remains constant after few iterations. It means that the state estimates never converges to its true state values. This is mainly due to the state observability issue. The relation between all state is not observed, i.e. the number of inter-mote distances obtained is less than the actual needed for each state to be observed completely. Now, assuming that the cricket motes are attached with omni-directional US transmitters and receivers, so that each mote can obtain distance measurements from adjacent motes.

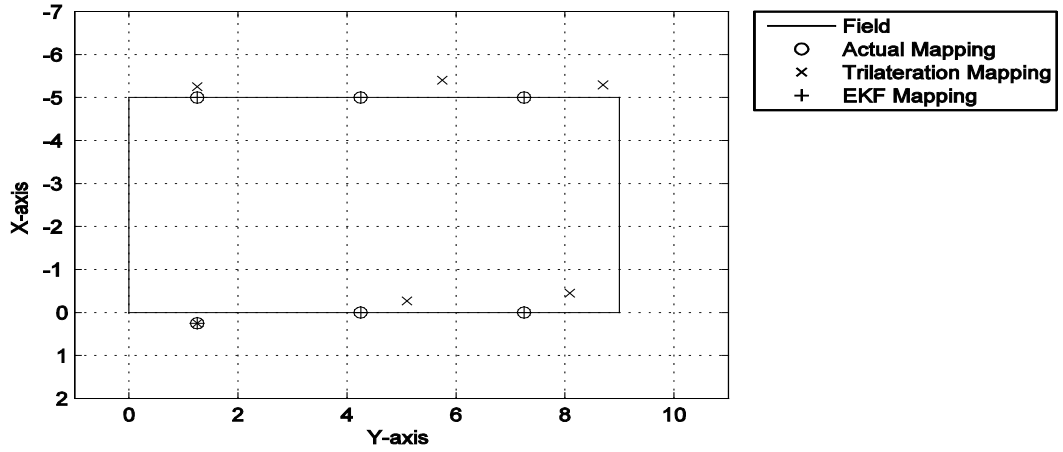

Figure 13. Mapping with low RMS error value in case of increased measurements.

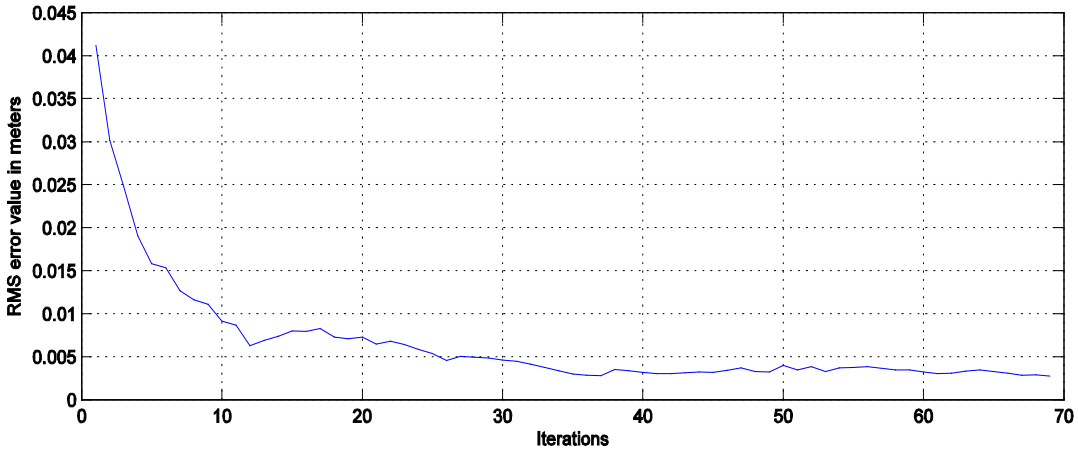

Figure 14. RMS error value for the state at each each iteration.

Figure 13 and 14 shows the case of mapping with low RMS error value (RMS error value is less than 5mm) when the number of distance measurements are increased. As shown in figure 14, the desired level of RMS error value is obtained in few iterations. Therefore, with increase in the number of measurements, the observability of the states increases and motes are mapped accurately. For a given network N_i, motes N^2_i and N^3_i know their distance and orientation with respect to the mote N^1_i. Using the knowledge of the mapped mote N^1_i (in phase 2), coordinates for motes N^2_i and N^3_i can be obtained. Motes in all networks are mapped accurately which can provide localization to an autonomous robot.

5. Conclusions

This paper presents a novel algorithm for self-mapping of the cricket motes. These self-mapping cricket motes can provide indoor mapping of a given location. The error in mapping due to trilateration can be eliminated using the SLAM algorithm. Using the EKF technique, the RMS error value is reduced.

Mapping with low RMS error value can only be obtained if large number of inter-mote distance measurements is available. Due to non omni-directionality of the US sensors of the cricket motes, only few inter-mote distance measurements can be obtained; hence accurate mapping cannot be obtained. The range of US sensors on cricket mote is limited (less than 10 meters) and thus the mapping for large fields cannot be obtained. Also, the numbers of motes used for mapping the shaded region are 18, which is still a costly factor.

Cricket mote cost around $10 non-commercially. Addition of extra US sensors (to make the US transmitter and receiver omnidirectional) will cost the whole unit to be less than $30. Making an omnidirectional US sensor will solve the state observability issue and will result in precise mapping. Also the range of the given US sensor can be increased up to 16 meters, which will provide mapping of larger areas. The algorithm can be easily extended to 3D localization.

REFERENCES

[1] http://www.wmccat.com/products/constTechGPS.jsp
[2] http://www.crssa.rutgers.edu/courses/gps/land_1_files/frame.htm
[3] Priyantha, N., B., Chakraborty, A., Balakrishnan, H., "The cricket location support system "Proc. 6th ACM MOBICOM, Boston, MA (2000).
[4] http://en.wikipedia.org/wiki/Trilateration
[5] Eren, T., Goldenberg, D., Whiteley, W., Yang, Y.,Morse, A., Anderson, B., Belhumeur, P., "Rigidity, computation, and randomization in network localization" IEEE INFOCOM (2004).
[6] Durrant-Whyte, H., Bailey, T., "Simultaneous Localisation and Mapping (SLAM):Part I The Essential Algorithms" Robotics and Automation Magazine, vol. 13 pp108, (2006).
[7] Julier, S., Uhlmann, J., "A new extension of Kalman Filter to nonlinear systems" SPIE AeroSense Symposium, 21–24 (1997).
[8] Bishop, G., Welch, G., "An introduction to the Kalman Filter," Course 8, Presented at ACM SIGGRAPH (2001).