Indoor Localization by using Particle Filtering Approach with Wireless Sensor Nodes

Hakan Koyuncu*, and Ahmet Çevik

Abstract : **Jennic type wireless sensor nodes are utilized together with a novel particle filtering technique for indoor localization. Target objects are localized with an accuracy of around 0.25 meters. The proposed technique introduces a new particle generation and distribution technique to improve current estimation of object positions. Particles are randomly distributed around the object in the sensing area within a circular strip of 2 STD of object distance measurements. Particle locations are related to object locations by using Gaussian weight distribution methods. Object distances from the transmitters are determined by using received RSSI values and ITU-R indoor propagation model. Measured object distances are used together with the particle distances from the transmitters to predict the object locations.**

Index terms: **Wireless sensor node (WSN), received signal strength indicator (RSSI), Gaussian weight distribution, Standard deviation (STD), particle filter, cumulative distribution**

I. Introduction

Wireless sensor nodes and localization algorithms are utilized together to determine the unknown object locations in indoors. These nodes are deployed in buildings with wireless sensor networks. Object localization such as surveillance, access control and location based identification are a few applications [1-3]. In most popular localization systems, objects carry active radio frequency identification (RFID) transmitters or receivers. Distance calculations can be carried out by using time of arrival (TOA) [4], or time difference of arrival (TDOA) [5], or received signal strength indication (RSSI) [6,7]. RSSI based localization techniques have the advantage of using inexpensive hardware.

Probabilistic state–space models are also important techniques in object localization by predicting the object positions with observed RSSI values [8,9]. If these models are linear and all the noise is additive Gaussian then classical Kalman filtering [10], technique is used. Another localization solution is given by the particle filtering which does not depend on any conditions of linearity or Gaussian noise [8,11].

Particle filtering is more robust for inexpensive indoor applications like RFID tracking systems. They are sequential Monte Carlo methods [12], based on point mass (or “particle”) representations of probability densities. They can be applied to any state-space model and generalize the traditional Kalman filtering methods.

Particle filter was first proposed by Kitagawa et al. in 1996, [18]. In the proposal each particle is randomly distributed and the trajectory of particles in successive prediction stages is simulated by using an assumed model. In the filtering stage, the weight proportional to likelihood is utilized to get the next set of particles which represented the filter distribution. Recently J.Tsuji et al. in 2010, [13] employed particle filtering for localizations with ZigBee devices. In this study, object locations are determined probabilistically by using previous object locations and their received RSSI values with randomly distributed particles.

In our study, a novel particle filtering technique is introduced to determine the unknown object locations by utilizing a distribution model of the random particles. In this technique, random particle generation is carried out in circular strips dependent on received RSSI values around the sensors instead of randomly distributed across the total test area as in literature.

Sequential estimation is used to increment the position of the particles as RSSI data is gathered during the process. RSSI data, arriving from the sensor nodes, is received at object location and then converted into position data. This object position data is later compared with the particle position data. The difference between two is utilized in a Gaussian probabilistic model to obtain the weights of the particles. Particles with smaller differences with measured object locations produced more weights and their likelihood of transferring to next sequential stage is increased. Each sequential stage corresponded to an object position measurement and identified as iteration. Weighted Particles are translated to next stage by incremental distances at every iteration and they get closer to object location. At the end of a number of iterations, particle averaging reveals an accurate object position.

II. Methodology

A. Particle Filtering Approach

An object location x_t can be calculated by using particle filtering technique [12], where x_t depends on previous location, $x_{t-1,}$ and it is obtained probabilistically. Received RSSI vector y_t= ($RSSI_1$,$RSSI_2$,…$RSSI_t$) is related to object

Manuscript received January 11, 2013 , revised February 17, 2013. This work has been supported by Computer Science Department of Loughborough University ,UK , in 2011

Authors are with Loughborough University,Loughborough, United Kingdom and Leeds University, Leeds ,United Kingdom. Emails : h.koyuncu@lboro.ac.uk , mmac@leeds.ac.uk

*Corresponding author

location x_t and it is also calculated probabilistically. Consequently, Particle filter technique estimates a distribution of x_t.

An object location, defined by a distribution of x_t ,can be described by a set of estimated locations identified as "particles" ,$\{p_k\}$, where k is the user defined particle numbers ($1 \leq k \leq N$). A particle p_k can have a location x_k and its weight can be w_k [13]. The weight of a particle can be identified by utilizing the minimum distance difference, Δx_k, between the measured object distances and the particle distances to transmitters.

A weight function which produces larger weights with smaller Δx_k values is required for particles to move to next iteration. Hence standard normal distribution function with standard deviation value of 1 is utilized as weight function with equation (1).

$$w_k = \frac{1}{\sqrt{2\pi R}} e^{-\frac{(\Delta x_k)^2}{2R^2}} \qquad (1)$$

where k is the particle index and R is the standard deviation and equal to 1.

w_k values of the particles are used to determine which particles will be employed in the current object location guess process. Particle weights are normalized with respect to total sum of particle weights and sum of normalized particle weights are equal to 1. See equation (2).

$$w_{norm\,k} = \frac{w_k}{\sum_{k=1}^{N} w_k} \qquad (2)$$

In each iteration, particles are incremented by a motion model and then corrected by the sensor measurement. Particles are generated randomly in this motion model. In the correction step, each particle is moved according to the incoming RSSI sensor data. If a particle is far from the measured object position with respect to receiver sensor then the particle weight is lowered, if the particle is close to measured object position then its weight is increased.

B. Cumulative Resampling

A cumulative distribution function [14], of normalized weights of the particles is generated between 0 and 1 to determine which particle must be carried in current object location estimation. See equation (3).

$$Cum_k = \sum_{i=1}^{k} w_{normi} \qquad (3)$$

Where k is particle index and $w_{norm\,i}$ is the normalized weight for the particle with index i.

Cumulative distribution function utilizes total number of particles and its x axis is particle index and its y axis is cumulatively normalized particle weights. Normalized weights are cumulatively added as the particle index is increased. The sum of all previous particles' normalized weights until current particle will be the cumulative normalized weight of the current particle defined by y coordinate and its particle index defined by x coordinate of cumulative distribution function. Since all the particles are included in this process, minimum values of the cumulative distribution function is 0 particle index and 0 cumulative normalized weight while maximum values of cumulative normalized weight is 1 and particle index is the total number of particles in current process.

Cumulatively normalized weighted particles will be transferred to next iteration with equal probability by assuming a random model. In this model, random floating point numbers between 0 and 1 are generated at equal numbers with particles. Any random number which is closest to any cumulative normalized weight will be selected and the corresponding particle index will be included in current object location guess process. During the selection process, particles with higher cumulative normalized weights are transferred to next iteration. Particles with smaller cumulative normalized weights will be ignored. Selected particles will be incremented to next iteration and included in next object location guess process. Coordinates of selected particles are averaged out to give the estimated current position of the object.

C. Particle Generation

A rectangular, obstacle free, sensing area is deployed in this study. It has 4 transmitter sensors at 4 vertexes of the rectangular area and a target object at T in **Fig. 1**. Particles are generated randomly between predetermined boundaries. RSSI values, arriving from the transmitter sensors, are measured at object point and converted into distance values by using ITU-R indoor propagation path loss model [15]. RSSI values, received by the receiver sensor on the object [16], can be expressed by this model as

$$RSSI = P + G + \alpha - \beta.\log(d) \qquad (4)$$

Where P is the transmission power, G is the receiver gain, α and β.log(d) are the amount of path losses according to ITU model. d is the distance between the object and the related transmitter sensor node across the sensing area.

In particle filtering technique, in literature, random particles are generated and distributed randomly across the total sensing area which resulted large number of iterations to localize the object. But in this study, Random particles are generated and randomly distributed within limited circular strip areas around each transmitter sensor node as shown in Fig. 1. A circular strip is generat(3) between the sensor area boundaries AD and AB and the circles with radiuses (dA_1 + dA_{STD}) and (dA_1 - dA_{STD}) for sensor A . dA_1 is the first dA measured object distance to transmitter node A and it is calculated with equation (4) by using RSSI measurements received from A. dA_{STD} is the STD value of all the calculated dA values from RSSI recordings at object location with respect to A transmitter.

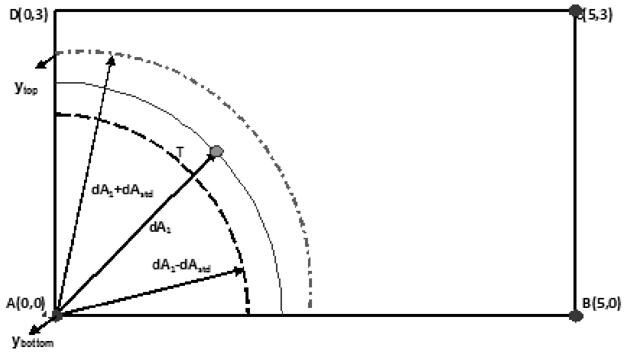

Fig. 1. Circular strip between $dA_1 + dA_{STD}$ and $dA_1 - dA_{STD}$ circles for 1st object distance dA_1 with transmitter sensor A.

Similar circular strips are also generated around the other transmitter sensors as seen in Fig. 2. Transmitter sensor coordinates will be utilized to determine the boundaries of the particles. Y coordinates of the random Particles will be between y_{top} and y_{bottom} as shown for transmitter sensor A in Fig. 1. Y coordinates of all the particles in different circular strips can be determined by using d distances between the measured object locations and transmitter nodes and their STD values.

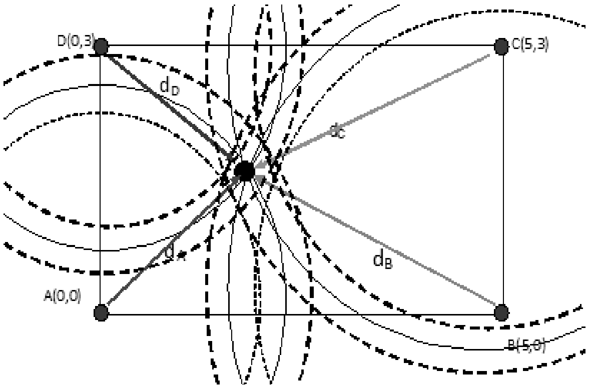

Fig.2. Circular particle strips with respect to sensors with coordinates $A(x_A,y_A)$, $B(x_B,y_B)$, $C(x_C,y_C)$, $D(x_D,y_D)$

D. Y_{top} and Y_{bottom} calculations for A, B, C and D sensors

Rectangular sensing area boundaries are defined by AB, BC, CD and AD where AB = CD and BC = AD. General sensor coordinates are deployed as $A(x_A,y_A)$, $B(x_B,y_B)$, $C(x_C,y_C)$, $D(x_D,y_D)$. Circular strips are generated with first dA, dB, dC and dD distances which are calculated for first received RSSI values. Top and Bottom values for dA, dB, dC and dD are defined as follows:

$dA_{bottom} = dA - dA_{STD}$, $dB_{bottom} = dB - dB_{STD}$, $dC_{bottom} = dC + dC_{STD}$, $dD_{bottom} = dD + dD_{STD}$ and $dA_{top} = dA + dA_{STD}$, $dB_{top} = dB + dB_{STD}$, $dC_{top} = dC - dC_{STD}$, $dD_{top} = dD – dD_{STD}$. Y_{top} and Y_{bottom} values are presented here with respect to sensing area boundaries and A and B sensor coordinates.

a) If $dA_{bottom} \leq AD$ or $dB_{bottom} \leq AD$, than ($Y_{bottom} = y_A$) for sensor A or ($Y_{bottom} = y_B$) for sensor B.

b) If $dA_{top} \leq AD$ or $dB_{top} \leq AD$ than ($Y_{top} = y_A + dA_{top}$) for sensor A or ($Y_{top} = y_B + dB_{top}$) for sensor B.

c) If $AD \leq dA_{bottom} \leq AB$ or $AD \leq dB_{bottom} \leq AB$, than ($Y_{bottom} = y_A$) for sensor A or ($Y_{bottom} = y_B$) for sensor B.

d) If $AD \leq dA_{top} \leq AB$ or $AD \leq dB_{top} \leq AB$, than ($Y_{top}=y_A + AD$) for sensor A or ($Y_{top} = y_B + BC$) for sensor B.

e) If $AB \leq dA_{bottom} \leq \sqrt{AB^2 + BC^2}$ or $AB \leq dB_{bottom} \leq \sqrt{AB^2 + BC^2}$ than ($Y_{bottom} = y_A + \sqrt{dA_{bottom}{}^2 - (x_B - x_A)^2}$) for sensor A or ($Y_{bottom} = y_B + \sqrt{dB_{bottom}{}^2 - (x_B - x_A)^2}$) for sensor B.

f) If $AB \leq dA_{top} \leq \sqrt{AB^2 + BC^2}$ or $AB \leq dB_{top} \leq \sqrt{AB^2 + BC^2}$ than ($Y_{top} = y_A + AD$) for sensor A or ($Y_{top} = y_B + BC$) for sensor B.

g) If $\sqrt{AB^2 + BC^2} \leq dA_{bottom}$ or $\sqrt{AB^2 + BC^2} \leq dB_{bottom}$ than there are no Y_{bottom} due to no intersection with the sensor area boundaries.

h) If $\sqrt{AB^2 + BC^2} \leq dA_{top}$ or $\sqrt{AB^2 + BC^2} \leq dB_{top}$ than ($Y_{top} = y_A + AD$) for sensor A or ($Y_{top} = y_B + BC$) for sensor B.

Similarly, Y_{top} and Y_{bottom} values with respect to sensing area boundaries and C and D sensor coordinates can also be determined by using symmetry properties with A and B sensors. Finally, they will all be deployed to calculate the coordinates of the random particles.

E. (x, y) coordinates of random particles

Y coordinates of the random particles can be expressed as (in MATLAB version 9) :

$$y = Rand(1\,,k) \times (y_{top} - y_{bottom}) + y_{bottom} \tag{5}$$

where k random numbers are generated between 0 and 1. A random circle with radius R_{Random} and circle centre $A(x_A , y_A)$ is generated between the circles with radiuses $dA_1 + dA_{STD}$ and $dA_1 - dA_{STD}$ for sensor A as an example. R_{Random} is given as

$$R_{Random} = Rand(1\,,\,k) \times (2 \times dA_{std}) + (dA_1 - dA_{std}) \tag{6}$$

The equation of the R_{Random} circle can be expressed as

$$(x - x_A)^2 + (y - y_A)^2 = R^2_{Random} \tag{7}$$

Hence the x coordinate of the same random particle, can be calculated as

$$x = \sqrt{R^2_{Random} - (y - y_A)^2} + x_A \tag{8}$$

R_{Random} and x,y coordinates are similarly calculated for B,C,D sensors as well.

F. New particle prediction mechanism

Initially, $\{P_k\}$ random particles are generated for each circular strip. There are 4 circular strips and (x, y) coordinates of the particles in these strips are calculated. The first estimated object position is identified as the mean values of all the particle coordinates as shown in equation (9).

$$(X_{1st-estimate}\,,\,Y_{1st-estimate}) = (\frac{1}{4k}\sum_{i=1}^{4k} x\,,\frac{1}{4k}\sum_{i=1}^{4k} y) \tag{9}$$

($X_{1st\text{-}estimate}$, $Y_{1st\text{-}estimate}$) coordinates are the first estimated coordinates of the object location. Object's first position estimation can converge to its second position estimation by deploying 2nd object distance measurements dA$_2$, dB$_2$, dC$_2$ and dD$_2$ as follows;

The difference between the 2nd and 1st object distance measurements with respect to sensors are expressed as (dA$_2$ - dA$_1$), (dB$_2$ - dB$_1$), (dC$_2$ - dC$_1$), (dD$_2$ - dD$_1$). The mean (S$_{MEAN}$), and standard deviation (S$_{STD}$) of these 4 differences are determined in order to quantize the particle incremental motion between iterations. Particle motion consists of a step distance L and a step angle ϴ. L step distance is related to S$_{STD}$ and S$_{MEAN}$ of two consecutive object distance measurements with respect to sensors, while ϴ angle is selected between 0 and 2π radian for each particle step. These quantities define the amount of movement as the particle converges from previous position to next position.
L and ϴ are defined as:

$$L = (Rand(1\,,k) \times 2 \times S_{STD}) + (S_{MEAN} - S_{STD}) \qquad (10)$$

$$\theta = (Rand(1,\ k) \times 2 \times \pi) \qquad (11)$$

As the particle at (x,y) position moves to a new position by the amounts of L andϴ , the new modified particle position coordinates become ;

$$x_{new} = x + L \times Cos(\theta^0) \qquad (12)$$

$$y_{new} = y + L \times Sin(\theta^0) \qquad (13)$$

Hence (x$_{new}$, y$_{new}$) coordinates of the particles are calculated by using received RSSI data and the resulting STD and MEAN values of distance measurements between iterations. Finally these particles are transferred to next iteration step.

Distance differences between the 2nd object distance measurements and the distances of new particles, (x_{new}, y_{new}), with respect to sensors are determined. Minimum distance differences are defined as Δx$_k$. During the second iteration, new particle weight calculations are utilized for the (x_{new}, y_{new}) particles by using minimum distance differences, Δx$_k$ and equation (1).

New particle weights are normalized with equation (2) and the cumulative distribution of normalized weights is determined with equation (3). Cumulatively normalized weights are compared with the randomly generated floating point numbers and closest particles with new coordinates are selected as $(x_{new+1}\ ,\ y_{new+1})$ and transferred to next iteration.

Second estimated object position is identified as the mean value of these particles with coordinates (x_{new+1}, y_{new+1}) and expressed as (X$_{2nd\text{- }estimate,}$ Y$_{2nd\text{- }estimate}$). A block diagram for operational phases is shown in Fig. 3.

Similar correction, modification and translation stages can be deployed for 3rd object position estimation and the new estimated object position is expressed as (X$_{3rd\text{-}estimate,}$ Y$_{3rd\text{-}estimate}$). This procedure continues until the total number of object position measurements are processed sequentially between iterations.

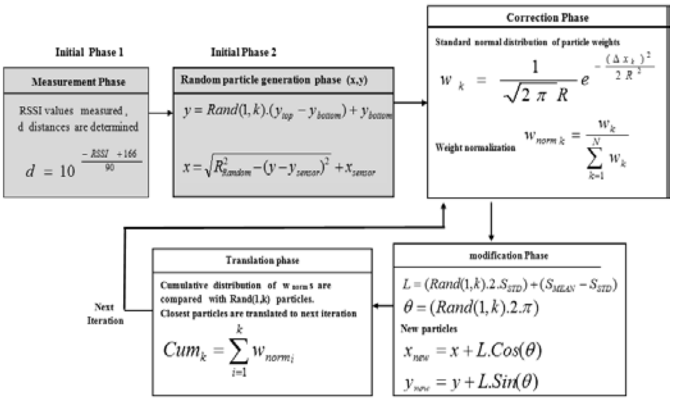
Initial Phase 1
Measurement Phase
RSSI values measured , d distances are determined
$d = 10^{\frac{-RSSI\ +166}{90}}$
Initial Phase 2
Random particle generation phase (x,y)
$y = Rand(1,k).(y_{top} - y_{bottom}) + y_{bottom}$
$x = \sqrt{R^2_{Random} - (y - y_{sensor})^2} + x_{sensor}$
Correction Phase
Standard normal distribution of particle weights
$w_k = \frac{1}{\sqrt{2\pi}\ R} e^{-\frac{(\Delta x_k)^2}{2R^2}}$
Weight normalization
$w_{norm\ k} = \frac{w_k}{\sum_{k=1}^{N} w_k}$
modification Phase
$L = (Rand(1,k).2.S_{STD}) + (S_{MEAN} - S_{STD})$
$\theta = (Rand(1,k).2.\pi)$
New particles
$x_{new} = x + L.Cos(\theta)$
$y_{new} = y + L.Sin(\theta)$
Translation phase
Cumulative distribution of w$_{norm}$s are compared with Rand(1,k) particles.
Closest particles are translated to next iteration
$Cum_k = \sum_{i=1}^{k} w_{norm_i}$
Next Iteration

Fig. 3. A general block diagram for operational phases (initial phases are highlighted)

III. IMPLEMENTATION

Sensing area where the measurements are carried out has dimensions of 5 m by 3 m in the middle of a basketball field and a grid space of 0.5 m as shown in Fig. 4. Measurement model, utilized here, does not include multipath, fading or shadowing effects due to small size and open boundaries of the test area. There are no obstacles across the sensing area and all the measurements are taken as line of sight measurements. Many tests are carried out and a large number of unknown object locations is estimated. A number of them is presented as an example in this paper.

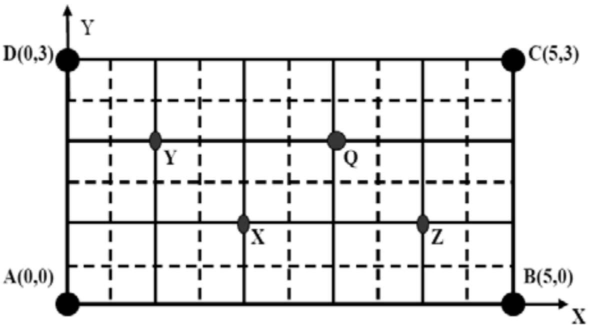

Fig. 4. 5mx3m sensing area with 4 transmitters A,B,C,D and 4 objects with receivers at X, Y, Z, Q locations

Jennic type transmitters and receivers, [17], are employed in the experiments. M = 100 LQI readings are received from each transmitter A,B,C,D by the receiver on the target at different locations. LQI values are converted into RSSI values by using LQI-to-RSSI Jennic conversion tables. Later on, RSSI values are converted to “d” distance values between the transmitters and receiver by using equation (4). All the RSSI data is collected by the receiver on the object and sent to a gateway server computer. All the computations are carried out with this server.

At any object location, a set of 4 *d* distance values, { d$_A$, d$_B$, d$_C$, d$_D$ }, are calculated by using received RSSI values M times. Each set corresponds to a single iteration and there is M number of maximum iterations. The MEAN and STD values of these M distances are calculated and utilized to develop the circular strips across the sensor area. User defined number of random particles within the circular boundaries are generated and object coordinates are estimated at every iteration by going through the operational stages in block diagram in Fig. 3.

Circular strips of random particles around each transmitter sensor and the estimated object position for the object at X(2,1) location are presented at the end of first iteration in Fig. 5a. As the particles move from first iteration to next iterations, particle positions and the estimated object positions are displayed in Fig. 5b to 5f. The system is also deployed to estimate the location of the objects by using standard particle filtering in literature where the particles are distributed randomly across the total test area as shown in Fig.5g .

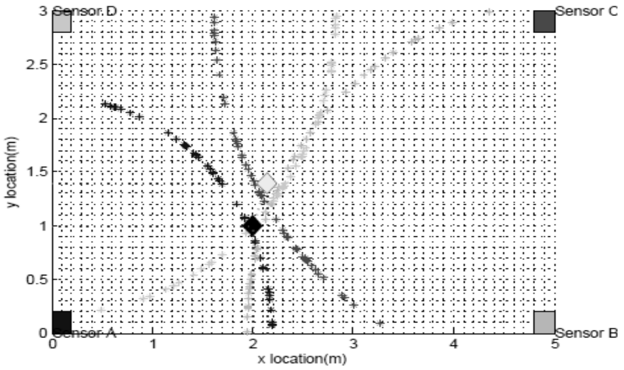

Fig. 5a. Iteration 1, object position (2,1) , estimated object position (2.2 , 1.4)

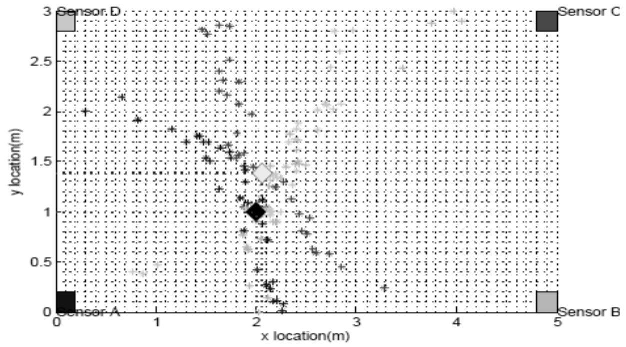

Fig.5b. Iteration 3, object position (2,1) , estimated object position (2.1 , 1.4)

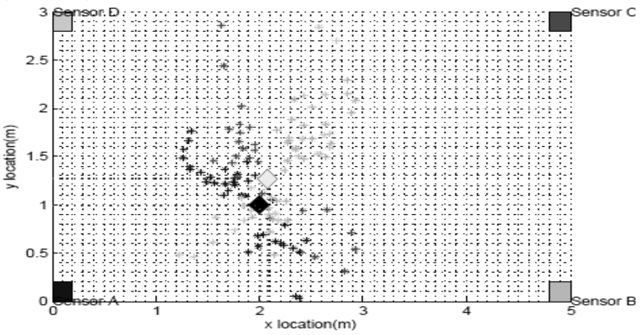

Fig.5c. Iteration 5, object position (2,1) , estimated object position (2.1 , 1.3)

10 random particles for each circular strip, totaling 40 particles for 4 strips, are utilized as an example to give an overview of particle transition between the iterations and to estimate the object location Y(1, 2). 10 Random particles are generated and their (x,y) coordinates are calculated for each circular strip by using equations (5),(6) and (8).

The mean value of 40 particle coordinates is defined by equation (9) as the first estimated object position of ($X_{1st\text{-}estimate,}$ $Y_{1st\text{-}estimate}$) = (1.551, 1.849).

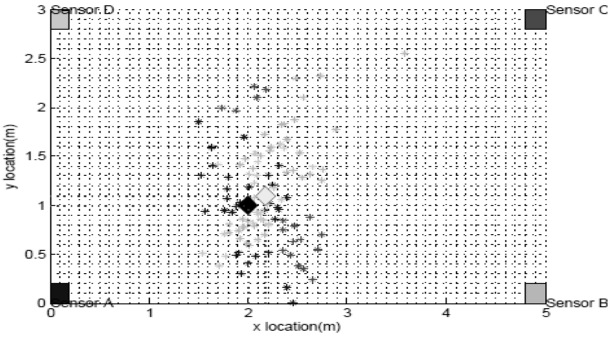

Fig. 5d. Iteration 10, object position (2,1) , estimated object position (2.2 , 1.1)

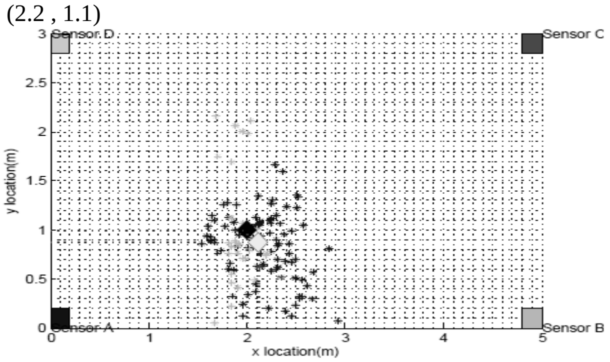

Fig. 5e. Iteration 30, object position (2,1), estimated object position (2.1 , 0.9)

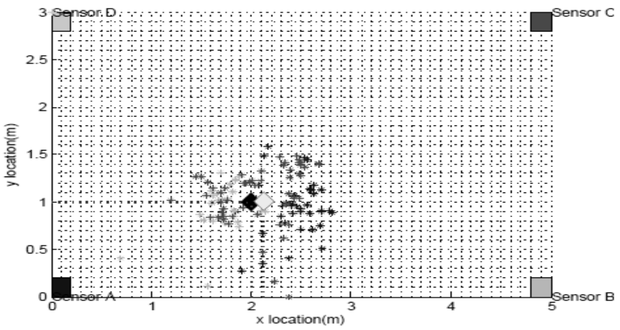

Fig.5f. Iteration 50, object position (2,1), estimated object position (2.1 , 1.05)

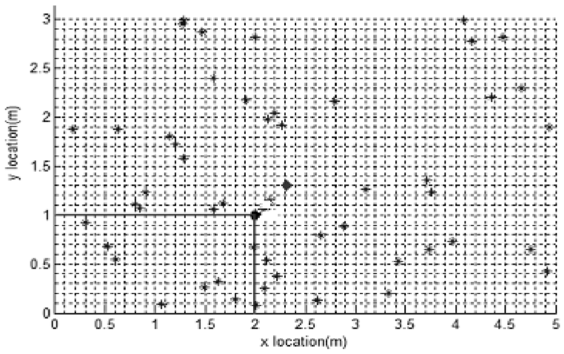

Fig.5g. Iteration 30, object position (2,1), estimated object position (2.3 , 1.4) , random particle distribution

Differences between the 2nd and 1st calculated object distances with respect to sensors, their mean (S_{MEAN}), and Standard deviation (S_{STD}) values are determined. They are utilized with (x,y) coordinates of initial random particles and Equations (10) and (11) to determine the x_{new} and y_{new} Coordinates for object location (1,2). See Table I.

These cumulative w_{norm} values are compared with 10 randomly generated numbers between 0 and 1. Particles with closest cumulative normalized weights to random numbers are selected as (x_{new+1},y_{new+1}) particles and they are incremented to second iteration as shown in Table I.

TABLE I

Random Particles (X , Y) , Translated Particles ($X_{new,}$ Y_{new}) To 1ST Iteration And Translated Particles (X_{new+1} , Y_{new+1}) To 2ND Iteration For Object Location (1,2)

Initial particle number	random particles generated		L & Θ included x , y		Xnew + 1	Ynew + 1	Translated particle numbers
	x	**y**	**xnew**	**ynew**			
1	1.2130	2.4150	1.385	2.017	0.9380	2.4680	4
2	1.1760	2.6850	1.597	2.402	1.3850	2.0170	1
3	2.3960	0.3760	2.749	0.738	1.2040	2.7090	9
4	0.5410	2.7080	0.938	2.468	1.2040	2.7090	9
5	1.4270	1.8740	1.096	1.511	1.0960	1.5110	5
6	2.5450	0.2890	2.898	0.537	1.0960	1.5110	5
7	2.5900	0.8250	2.509	1.275	2.7490	0.7980	3
8	2.3750	1.6210	1.895	1.475	1.2040	2.7090	9
9	0.0730	2.8380	1.204	2.709	0.9380	2.4680	4
10	0.5160	2.8600	1.010	2.750	1.3850	2.0170	1

The distances between (x_{new} , y_{new}) particles and transmitter sensors A , B , C , D are calculated and the differences between these distances and the 2nd object distances dA_2 , dB_2 , dC_2 , dD_2 to sensors are deployed . The minimum of these 4 differences is termed as "min diff" for each particle. Particle weights are calculated by using these minimum differences and they are normalized to give w_{norm} values as shown in Table II for 10 particles. Cumulative distribution of w_{norm} values for 10 particles is shown graphically in Fig. 6.

TABLE II

Parameters Used Between 1ST And 2ND Iteration For Object Location (1,2)

particle	min diff	w weight	Norm weight w_{norm}	Cumulative w_{norm}	random numbers
1	0.280	0.383	0.1243	0.1244	0.4173
2	0.27	0.38	0.124	0.249	0.049
3	1.10	0.21	0.070	0.319	0.902
4	0.75	0.30	0.097	0.416	0.944
5	0.12	0.39	0.128	0.545	0.490
6	1.20	0.19	0.062	0.607	0.489
7	1.065	0.22	0.0730	0.6809	0.3377
8	0.452	0.36	0.1160	0.7977	0.9001
9	0.740	0.303	0.0980	0.8961	0.3692
10	0.662	0.320	0.1039	1.0000	0.1112

Particle 4 is highlighted to trace the conversion stages as an example of particle transfer to next iteration in Table I . Hence the second estimate of object position coordinates are calculated as the mean value of (x_{new+1} , y_{new+1}) coordinates in Table I and calculated as ($x_{2nd\text{-}estimate}$, $y_{2nd\text{-}estimate}$) = (1.32 , 2.086) .

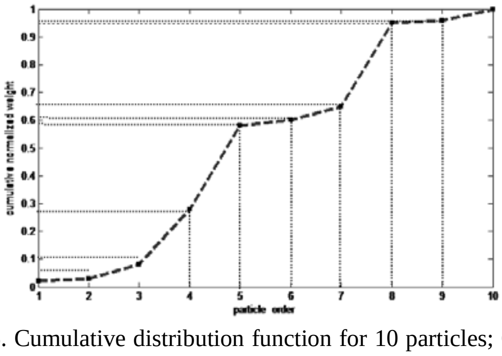

Fig. 6. Cumulative distribution function for 10 particles; Cumulative normalized weights is (y axis), particle index is (x axis).

Similar procedures are repeated and 3rd estimate of object location coordinates are determined as ($X_{3rd\text{-}estimate,}$ $Y_{3rd\text{-}estimate}$) = (1.165, 1.986). In conclusion, 10 particles are generated with equal probability for each circular strip at first iteration. At the end of first iteration, only particles with numbers 1, 3, 4, 5, 9 are translated to second iteration as shown in Table I.

Every iteration, the number of independent particles is decreased and same particles appear more and more. This process in return gets the particles closer to each other and produces an estimated object position closer to actual object position. Object and estimated object coordinates for different number of iterations are presented in Fig. 7 and Fig. 8.

Average of estimated object coordinates with 20 and 50 iterations are calculated and the error distances between the actual and estimated object locations are determined by using the following equation

$$e = \sqrt{(x_{obj} - x_{est.obj})^2 + (y_{obj} - y_{est.obj})^2} \qquad (14)$$

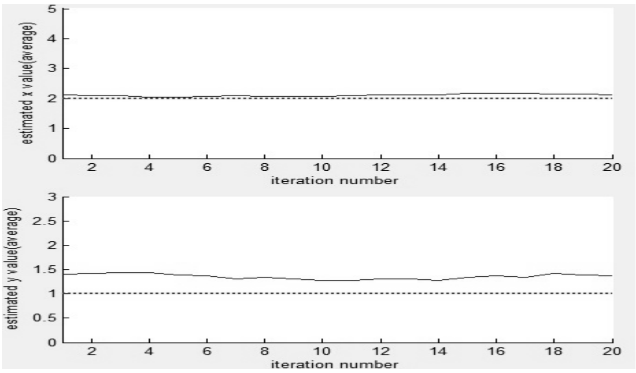

Fig. 7a. Object coordinates (2,1) and estimated object coordinates for 20 iterations

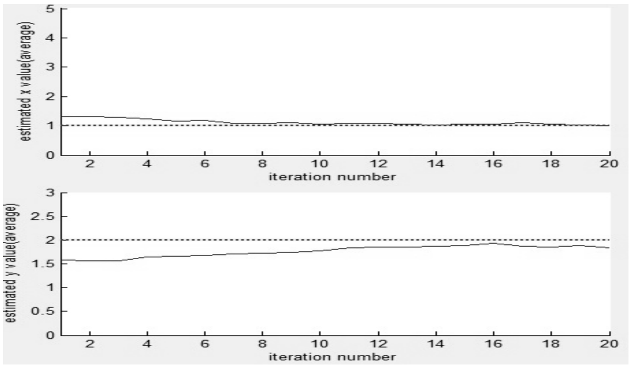

Fig. 7b. Object coordinates (1,2) and estimated object coordinates for 20 iterations

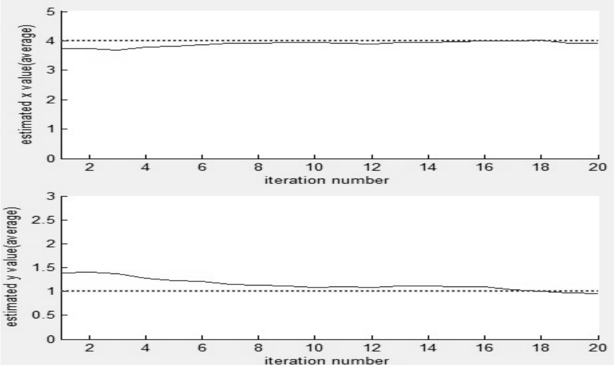

Fig. 7c. Object coordinates (4,1) and estimated object coordinates for 20 iterations

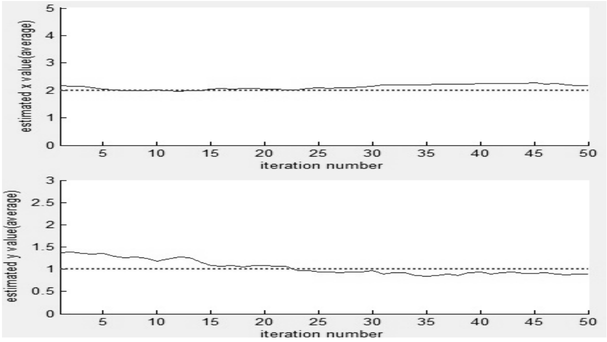

Fig. 8a. Object coordinates (2,1) and estimated object coordinates for 50 iterations

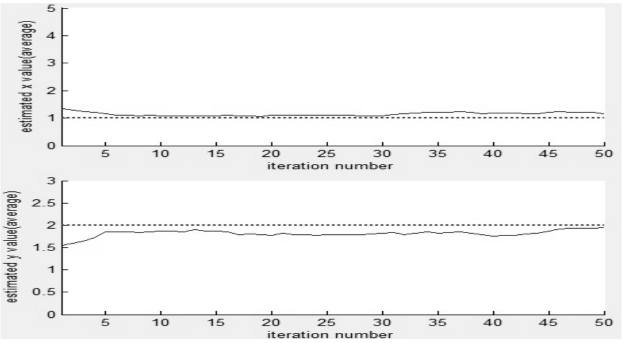

Fig. 8b. Object coordinates (1,2) and estimated object coordinates for 50 iterations

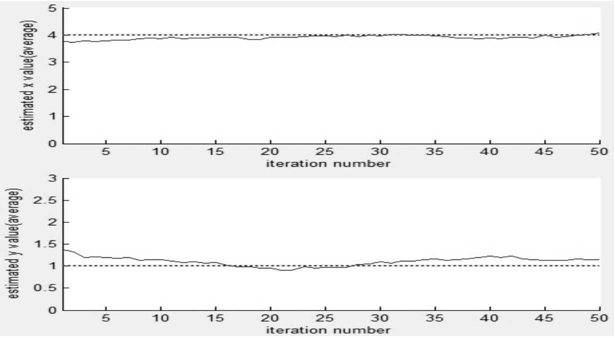

Fig. 8c. Object coordinates (4,1) and estimated object coordinates for 50 iterations

Table III and Table IV summarize the estimated object coordinates and the error distances between the object and estimated object coordinates for circularly distributed and randomly distributed particles.

Additionally these error distances are presented in bar chart form in Figure 9. It is clearly seen that the circularly distributed particles give less error distances compared to randomly distributed particles with particle filtering.

TABLE III

ACTUAL AND ESTIMATED OBJECT COORDINATES WITH PARTICLE FILTERING FOR CIRCULAR AND RANDOM PARTICLE DISTRIBUTIONS FOR 20 ITERATIONS

Object coordinate (x,y)	Estimated obj coordinates with circular particles	Error (m) (20iterations) (circular)	Estimated obj coordinates with Random particles	Errors (m) (20iterations) (Random)
2 , 1	2.109 , 1.353	0.37m	2.300 , 1,400	0.50m
1 , 2	1.118 , 1.766	0.26m	1.253 , 2.280	0.38m
3 , 2	2.913 , 1.654	0.35m	3.231 , 2.321	0.39m
4 , 1	3.894 , 1.147	0.18m	4.122 , 1.242	0.27m
1 , 1	1.200 ,1.250	0.32m	1.352 , 0.761	0.43m
2 , 2	2.242, 2.262	0.36m	2.328 , 1.656	0.48m
3 , 3	3.241 , 3.233	0.34m	2.752 , 3.431	0.50m
4 , 2	4.232 , 2.121	0.26m	4.241 , 2.223	0.33m
2 , 3	2.172 , 3.252	0.30m	2.256 , 3.261	0.37m
	Ave Error	0.30m	Ave Error	0.40m

TABLE IV

ACTUAL AND ESTIMATED OBJECT COORDINATES WITH PARTICLE FILTERING FOR CIRCULAR AND RANDOM PARTICLE DISTRIBUTIONS FOR 50 ITERATIONS

Object coordinate (x,y)	Estimated obj coordinates with circularparticles	Error (m) (50iterations) (circular)	Estimated obj coordinates with Random particles	Errors (m) (50iterations) (Random)
2 , 1	2.121 , 1.041	0.13m	2.221 , 1.250	0.33m
1 , 2	1.148 , 1.816	0.23m	1.241 , 2.253	0.35m
3 , 2	2.882 , 1.743	0.28m	3.257 , 2.263	0.37m
4 , 1	3.913 , 1.107	0.13m	4.234 , 0.841	0.28m
1,1	1.25 , 0.95	0.25m	1.183 , 0.733	0.32m
2,2	2.16 , 1.93	0.17m	2.241 , 1.800	0.31m
3,3	3.224 , 3.131	0.26m	3.161 , 2.752	0.29m
4,2	4.221 , 2.211	0.30m	4.282 , 2.271	0.39m
2,3	2.142 , 3.162	0.22m	2.251 , 2.742	0.36m
	Ave Error	0.21m	Ave Error	0.33m

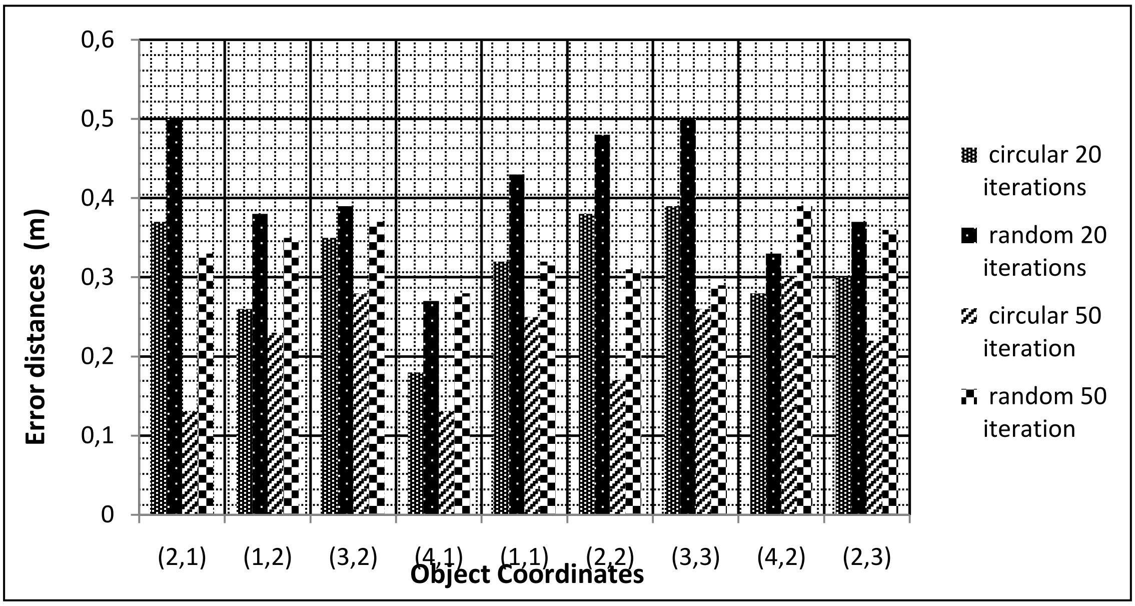

Fig. 9. Error distances at different object points for 20 and 50 iterations

IV. Conclusions

Unknown object locations are determined by using a novel implementation method introduced with particle filtering. The new method deploys a new technique in generation and distribution of the particles and estimates the object coordinates in time domain.

Secondly, ITU-R indoor propagation model is used in this study due to RF frequency ranges utilized with Jennic wireless sensors. Due to small size of the sensing area in a large open space of indoors, affects such as multipath, fast fading, shadowing or slow fading and RF noise are not included in the calculations. The measurements are taken as line of sight measurements and there are no obstacles across the sensing area.

Random particles are generated in circular strip areas which are defined by the area differences between the circles with radiuses d_{SENSOR} ± d_{STD} instead of randomly generated across the sensing area as in literature. Hence less number of iterations is employed to achieve higher accuracies in object localization.

d_{SENSOR} is the distance of the object to each transmitter sensor calculated by deploying received RSSI values with ITU propagation model. d_{STD} is the standard deviation of these distances . Upper and lower limits of these circles are the boundaries of the sensing area.

Selective Particles are incremented between every iteration depending on their weights and their cumulative weight distributions. New particle positions are predicted by adding finite distances depending on the received RSSI values between subsequent iterations to their previous position coordinates.

Particles with larger weights are carried to current guess process more than one. If the particle weight is small this particle is not carried to current guess process and it is ignored. This process is called correction in original particle positions. The process continues with particles which have larger weights hence closer distances to estimated object positions at each iteration. At the end of each iteration, the estimated object position is calculated as the mean value of the weighted particle coordinates which are incremented to next iteration.

An overall average localization error of around 25 cm is achieved by applying the proposed distribution method of particles. Classical particle filtering with random distribution is compared with the new technique as shown in Fig.5g. The results show in Table IV that the new technique has a better positioning accuracy compared to classical particle filtering. The localization experiments are repeated for many unknown object locations. In each case the positioning accuracies were around 20-30cm and better than the results with classical particle filtering with random particle distribution.

Finally, the newly developed technique can be deployed in obstacle free parts of indoors to determine the unknown object locations very accurately. Further research will be carried out in indoors with obstacles and localization algorithms will be developed to reduce errors further.

References

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, N. S. Correal : *Locating the nodes cooperative localization in wireless sensor networks*, IEEE signal Processing Mag,Vol. 22, pp. 54-69 (2005).

[2] A. Boukerche, H. A. Oliveria, E. F. Nakamura, A. Loureio: *Localization system for Wireless sensor networks*, IEEE wireless communication Mag ,Vol. 14, pp. 6-12 (2007).

[3] H. Liu, H. Darabi, P. Banerjee, J. Liu: *Survey of wireless indoor positioning techniques and system*, IEEE transactions on systems, man and cybernetics, Vol. 37 ,pp 1067 – 1080, (2007).

[4] Zaruba G. V., Huber M., Kamangar F. A., Chlamtac I.: *Indoor location tracking using RSSI readings from single Wi-Fi access point*, Wireless Networks, Vol 13, pp 221- 235 (2007).

[5] Jan Blumenthal, F. Reichenbach, D. Timmermann: *Position estimation in Ad hoc wireless sensor networks with low complexity*, Joint 2nd workshop on positioning navigation and communication, pp. 41-49, (2005).

[6] Yunhao Liu, Cho Lau, Paul A. P. Landmarc: *Indoor location sensing using active RFID*, pervasive computing and Communication, Proceedings of 1st IEEE int. conference, pp. 407- 415 (2003).

[7] D. Zang, J. Ma, Q Chen, L. M. Ni: *An RF based system for tracking transceiver free objects*, Proceedings of percom pp. 203-207 , (2007).

[8] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp: *A tutorial on particle filters for Online nonlinear/ on-gaussian Bayesian tracking*, IEEE trans. Signal processing, Vol. 50, pp.174-188 , (2002).

[9] X. R. Li ,V. P. Jilkov: *Survey of maneuvering target tracking, dynamic models*, IEEE trans., aerospace, electron systems, Vol. 39, pp. 1333- 1364, 2003.

[10] G.Binazzi , L.Chisci,F.Chiti, R.Fantacci, S.Menci: *Localization of a swarm of mobile agents via unscented Kalman filtering*, Proc. IEEE Int conf. of Communication .ICC, Germany, (2009).

[11] Kung-Chung Lee, A Oka,Emmanuel Pollakis, Lutz Lampe: *A comparison between unscented Kalman Filtering and particle filtering for RSSI-based tracking*, 7th workshop on positioning and communication, pp. 157 – 163 (2010).

[12] A.Doucet, N.Freitas, N.Gordon : *Sequential Monte Carlo in practic*e, Springer-Verlag, (2001).

[13] J. Tsuji, H. Kawamura, K. Suzuki, A. Sashima, K. Kurumatani: *Zigbee based indoor Localization With particle filter estimation*, IEEE, ISNN 978-1-4244-6587, pp. 1115-1120, (2010).

[14] Gentle J. E.: *Computational statistics*, Springer (2010).

[15] ITU-R Criteria for propagation data and prediction methods for the planning of indoor radio Communication systems and ratio local area networks in the frequency range 900MHZ to 100GHZ, ITU-R Rec., pp. 1238-1240 (1999).

[16] Wen-Jiang Feng, Xiao-Wei Bi : *A novel adaptive cooperative Location Algorithm for Wireless Sensor Networks*, International Journal of Automation and Computing, IJAC, No: 04-039.R2, (2011)

[17] http://www..jennic.com/jn5139_modules , JN5139 Wireless micro Controller Modules, (2012) .

[18] Kitagawa, G.: *Monte Carlo filter and smoother for non-Gaussian non-linear state space Models*, Journal of computational and graphical statistics, Vol. 5, No. 1, pp. 1-25, (1996).

Hakan Koyuncu is in computer science department of Loughborough University, UK. He is currently doing PHD research in the area of RFIDs and position detection of objects in indoors. His research interests include wireless sensors, wireless networks and mobile Communication. He has completed his MSC in multimedia and wireless networks. He has developed algorithms to determine the accurate positions of the mobile and stationary objects.

Ahmet Cevik is a research scientist in Mathematics Department of Leeds University. He completed his BSC and MSC in computer engineering departments. His research interests are in computational logic and computational theory. He is also involved in RFID signal databases and calculating the object positions by using computational algorithms.