Simulation Based Comparison of Geo-Location Methods in Wireless Networks

E. Balarastaghi
Department of Electrical Engineering, College of Engineering, Boroujerd Branch, Boroujerd, Iran
E-mail: iut-iman@yahoo.com

MR. Amini, A. Mirzavandi
Department of Electrical Engineering, College of Engineering, Boroujerd Branch, Boroujerd, Iran
E-mail: iman.amini@gmail.com, asramirzavandi@gmail.com

Abstract—There are many Geo-Location techniques proposed in cellular networks. They are mainly classified based on the parameters used to extract location information. In this study it is tried to have a new look to these positioning methods and to classify them differently regardless of parameters type. We classified these techniques base on mathematical algorithms which is used to derive location information of users in the network. Such algorithms are divided into three main subclasses in here, estimation theory based (MUSIC, ESPRIT), Meta-heuristic (Genetic, PSO...) and filtering approaches (Kalman, Particle, Grid, MH ). The proofs and details of how to apply techniques are presented and the simulation results are given.

Index Terms—Geo-Location, Cellular Network, Kalman Filter, Particle Filter, Metropolis Hastings, Estimation

I. INTRODUCTION

Day by day developing of wireless communication systems and cellular networks along with increasing demands for location based services make policy makers and implementers of wireless networks to improve and develop positioning techniques and algorithms in such networks. In addition FFC1 regulations such as E-9112 and emergency aids help to make this process faster and faster[1]. These aids can help users efficiently by ALI3. Therefore communication companies are looking for corresponding methods and algorithms that both more accurate and cheaper and easier to implement on hardware.

Mainly there are two methods for user Geo-location. The first help the use of GPS devices called GPS aided. Each has its own advantages and disadvantages, for example for indoor Geo-location GPS cannot be useful. Replacing the role of satellites in the GPS service by that of base stations, the second method appears. This method uses the signals exist inherently in underlying communication network and tries to estimate user position by observing parameters that are dependent to location. This method can furthermore divides in two main approaches; mobile based and network based. The difference is the roles of mobile station and BTS in measuring the signals and estimating/calculating the location. In this paper we focus on mobile based techniques.

To determine the user location, two main steps should be passed, the first is measuring some signals and estimating the location dependent parameter or parameters from them. The second step is to use an algorithm to calculate (better to say again estimating ) the position. Many texts including thesis and book chapters discussed different methods of first steps [2,3,4,5,6,7,8]. These methods exploit measuring of parameters such as TOA4, TDOA5, AOA6, RSS7, DCM8, TA9 and other hybrid models. The second step may help the use of many algorithm such as Filtering methods , estimation based method and meta heuristic and evolutionary algorithms and some other approaches which are discussed here. Because of different simulation methods and scenarios it is not possible to compare these algorithms just by looking at the results of papers. They should be compared under unique simulation environment so we decided to have a performance evaluation between them. In this paper, Geo-location methods are classified based on mathematical algorithms used to find user location in the both area of deterministic and stochastic approaches, these algorithms can then use with the same parameter so comparison between them is rational and possible. Such algorithms divided into 3 main subclasses, estimation theory based (MUSIC, ESPRIT), Meta-heuristic (Genetic, PSO, ...) and filtering approaches (Kalman, Particle, Grid, MH ).

The contribution of this paper is to 1). Classifying Geo-location techniques based on mathematical approaches 2). Applying Simulation to them in the same environment 3). Comparing them based on both Radial Error and computational Complexity for implementation issues on hardware.

1 Federal Communication Commission
2 Enhanced 911
3 Automatic Location Identification
4 Time of arrival
5 Time difference of arrival
6 Angle of arrival
7 Received signal strength
8 Data correlation method
9 Timing advanced
The rest of this paper is organized as follow: section 2 describes the methods based on Estimation theory, in section 3, Meta-heuristic algorithms are presented and section 4 gives the information and different classes of Bayesian filtering in positioning. Finally in section 5 simulation details are described and some conclusions are given in section 6.

II. ESTIMATION THEORY BASED LOCALIZATION METHODS

Many algorithms in the class of estimation theory have been developed for location estimation. The most fundamental algorithms is based on Maximum Likelihood and Maximum A posterior Probability techniques and the their derivatives. MUSIC and ESPRIT\(^*\) are two of more popular. They use for so called ”super resolution ” estimation. Many variants of the two including fast versions, MIMO versions , adaptive and smart ones are applied to location estimation for estimating TOA,DOA\(^*\) and other parameters [5, 9, 10, 11, 12, 13, 14]. Both of them are essentially for estimating the frequencies of the signals. For example the first one, MUSIC, is based on linear algebra and vector space. If it is supposed that a signal is consist of desired signal and additive noise and composed of L single tones:

\[
u(n) = \sum_{l=1}^{L} a_l e^{j\omega_l n} z(n)
\]

(1)

Then the covariance matrix of signal samples, R, can be decomposed in the following way [6]:

\[
R = \sum_{\omega} S^{H} + \sigma^2 I
\]

(2)

Where S is the frequency matrix, D is the diagonal Correlation matrix of single tones and \(\sigma^2\) is the noise variance. They can be written in the following way:

\[
S = [s_1, s_2, ..., s_L]
\]

\[
D = \text{diag} \left( P_1, P_2, ..., P_L \right)
\]

(3)

It can be shown that Eigen values of matrix R can be divided in to two subspace, signal plus noise and noise as described below:

\[
\lambda_i = \begin{cases} 
\mu_i + \sigma^2 i; & i = 1, 2, ..., L \\
\sigma^2 i; & i = L + 1, ..., M 
\end{cases}
\]

(4)

Where \(\lambda_i\) and \(\mu_i\) are respectively the Eigen values of matrix R and SDS\(^H\) ordering in the ascending manner. Because all of Eigen vectors of a matrix form a perpendicular space, the two subspace must be orthogonal to each other. So the MUSIC algorithm is looking for a frequencies that are in the signal plus noise subspace and are orthogonal to the noise subspace. So the frequency sweep matrix, S(\(\omega\)), are defined to ensure which frequencies have the mentioned property. Thus the MUSIC Spectrum can easily be defined as:

\[
S_{\text{MUSIC}} (\omega) = \frac{1}{\mathbf{V}_N^H \mathbf{V}_N} \mathbf{r}^H (\omega)
\]

(5)

Where \(\mathbf{V}_N\) is the noise subspace vector. Each frequency having the peak value in the spectrum is in the signal frequency matrix. Some other versions such as Root-MUSIC and so on are also derived. By reciprocating the role of time and frequency one can easily apply these algorithm for TOA or in the same manner for DOA and so forth.

III. META-HEURISTIC LOCALIZATION METHODS

There are also many papers focusing on meta-heuristic and evolutionary algorithms to solve the final positioning equation. Their main idea is the same, defining the objective or fitness function and solving the optimization (min/max) problem by the corresponding algorithm operators [15, 16, 17, 18, 19, 20].

Mainly the objective function is defined based on the mean square error of population fitness and observed one as:

\[
F = \min \left( \sum_{i=2}^{\text{Cell}_N \text{No}} \sqrt{\text{SSE}} \right)
\]

(6)

For example in the case of TDOA, the difference distance of user location from reference BTS and the other is calculated as observed fitness and compared to that of for population generated by the algorithm.

\[
\text{SSE} = \left( x_{c.r,i,1} - x_{\text{Real}_\text{Coord}^{i,1}} \right)^2 + \left( y_{c.r,i,1} - y_{\text{Real}_\text{Coord}^{i,1}} \right)^2
\]

(7)

Where \(x_{c.r,i,1}\) is a difference distance between current population and reference ( indexed here by 1) and ith base station on x-axis and \(x_{\text{Real}_\text{Coord}^{i,1}}\) is a real difference distance (computed from measured TDOA values) of user and those base stations on x axis (so as \(y_{c.r,i,1}\) and \(y_{\text{Real}_\text{Coord}^{i,1}}\) on y-axis). So the algorithm starts to find the best point that is as close as possible and enough to the collision point of mentioned hyperbolic curves.(note that

---

\(^*\) Multiple signal classifier
\(^*\) Estimation of Signal Parameters via Rotational Invariance Technique
\(^*\) Direction of arrival

Copyright © 2015 MECS

I.J. Information Technology and Computer Science, 2015, 02, 44-53
in the case of TDOA, the final positioning equation leads
to intersection of two or more hyperbolic function. The
operators in GA\textsuperscript{13} for example is selection, mutation and
cross over to generate new generation. In the PSO\textsuperscript{14} there
are selection, mutation and position to be updated and help
converge the solutions. What is really obvious is that
none of these algorithms is suitable for online positioning
systems, so they are not considered so much.

IV. BAYESIAN FILTERING METHODS

There are different problems can be considered in the
field of estimation [21], namely:

The prediction problem, concerned with the
determination of

\[ \pi( X_k | Z_{t:k-1} ) \]

The filtering problem, concerned with the
determination of

\[ \pi( X_k | Z_{t:1} ) \]

The fixed-lag smoothing problem, concerned with the
determination of

\[ \pi( X_k | Z_{t:k+p} ), \quad P > 0 \]

where \( X_k \) is the State vector in the time instant \( k \) and
\( Z_{t:1} \) is all the measurements from time instant 1 to \( k \). It
is supposed to exist two equations or models: state update
(evolution) model (equation):

\[ X_k = f_k ( X_{k-1} ) + \text{process\_noise} \]

and measurement model (equation):

\[ Z_k = h_k ( X_k ) + \text{measurement\_noise} \]

By using the Bayes formula the process of filtering can
be decomposed into two steps: 1- State prediction
sometimes called prediction 2- measurement update
sometimes called correction. These steps are well
illustrated in the Fig.1.

Bayesian filtering is a technique that is used for
dynamic state estimating of a system with the help of
some parameters observed in the presence of noise called
noisy measurements. When a system is time variant many
estimation algorithms cannot work well [22, 23].

If we define the Belief as below:

\[ Bel( X_k ) = \pi( X_k | Z_{t:k} ) \]

then the two mention steps are mathematically derived in
this manner:

1- prediction step:

\[ Bel^\pi( X_k ) = \int \pi( X_k | x_{k-1} ) Bel( x_{k-1} ) dx_{k-1} \]

2- measurement update :

\[ Bel( x_k ) = C_k ? ( Z_k - x_k )? Bel^? ( x_k ) \]

The prediction step is derived by applying the
conditional and marginal probability theory:

\[ Bel^\pi( X_k ) = \pi( X_k | Z_{t:k-1} ) \]

\[ = \int \pi( X_k | x_{k-1}, Z_{t:k-1} ) \pi( x_{k-1} | Z_{t:k-1} )dx_{k-1} \]

Where the first and second term in the integral are
equal to \( \pi( X_k | x_{k-1} ) \) and \( Bel( x_{k-1} ) \) respectively
because \( x_{k-1} \) and \( Z_{t:k-1} \) has the same information due to
the measurement equation) measurement update step is
also derived by the Bayes rule. Note that the first step
practically calculates (estimates) the state variables
regarding behavioral action of state variables that state
equation shows (regardless of incoming measurement and
the second step updates and corrects the last values by the
current measurement observed.

In this concept and in the case of localization and also
other applications such as tracking [24,25,26], 3 popular
methods are well discussed in the literature, Kalman filter
( and its derivatives EKF\textsuperscript{15}, UKF\textsuperscript{16}\)), Particle filter
and Grid based method. Usually the EKF, UKF, Particle filter
and Grid algorithm are classified in the approximation
methods. Another classification is divided these methods
in two categories continuous and discrete. Based on this,
Kalman, EKF and UKF are continuous and Particle filter
and Grid algorithm are discrete.

\textsuperscript{13} Genetic Algorithm
\textsuperscript{14} Particle Swarm Optimization
\textsuperscript{15} Extended kalman filter
\textsuperscript{16} unscented kalman filter
A. Kalman Filtering

If the state and measurement equations are linear correspond to state variables (f and h are linear functions) and the noise (process and measurement) is assumed to have normal distribution, Bayesian filtering converts to Kalman filtering [7, 27] with the solution derived by Kalman. In this case the two equations can be written in a matrix form. If the equations are not linear, then to have the previous solutions for the problem, first and second order of Taylor series approximations are used instead. In this case the corresponding Jacobean matrix is replaced. That is why these methods are called approximated.

For Kalman filtering to become an efficient estimator for localization problems, the dynamic state equation must be modeled accurately. There are many models developed in this case. Some of them are deterministic and the others are stochastic. Deterministic models exploit the Newton motion models (position, velocity and acceleration equations) and simple and efficient to use.[27, 28]. For example Newton equations states that:

\[
X = X_0 + VΔt + \frac{1}{2} aΔt^2 \\
V = V_0 + aΔt \\
a(t) = a
\]

Hence the state variables could be position and velocity and acceleration in both x and y directions (in 2D scenario). Thus the matrix form of state equation is:

\[
X_k = AX_{k-1} + noise
\]

Where

\[
X_k = \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \\ a_{x,k} \\ a_{y,k} \end{bmatrix}
\]

and

\[
A = \begin{bmatrix} 1 & 0 & Δt & 0 & \frac{1}{2} Δt^2 & 0 \\ 0 & 1 & 0 & Δt & 0 & \frac{1}{2} Δt^2 \\ 0 & 0 & 1 & 0 & Δt & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}
\]

\[
X_k\] is the state vector and A is the coefficient matrix of state equation. Note that if each of velocity and acceleration terms does not come in the state vector as state variable, their impact should be modeled in the process noise.

Stochastic methods are usually use the Stochastic processes such as Random walk, Brownian motion and wiener process. These are more complex than the previous models [27, 28].

Measurement equation in the Kalman filtering depends on the observed parameters in the cellular network which can be TOA, RSS, TDOA, and so on discussed before. This equation could be either linear (in the case of TOA as an example) or nonlinear (in the case of RSS for example) so for the former case the problem is the exact Kalman filtering and for the latter case EKF could be used to solve. As an example of EKF, in the case of RSS, measurement models exploit different propagation path loss models along with shadowing effect[29, 30, 31, 32]. One simple model could be written in the form:

\[
Z_{k,i} = Z_{0,i} - 10α log_{10}(d_{k,i}) + ν_{k,i}
\]  \( (8) \)

Where α is the path loss exponent (PLE), \( Z_{k,i} \) is the received power from the \( i \)th BTS at time \( k \), \( d_{k,i} \) is the distance between mobile station and \( i \)th BTS and \( ν_{k,i} \) is the noise value in the \( i \)th link that is called shadowing and has log-normal distribution. It is completely nonlinear and must be approximated by Taylor series to be used by kalman technique called EKF. In this case The matrix form, the coefficient matrix of measurement equation achieved based on Jacobean definition discussed earlier:

\[
H = 1n(10)*G
\]  \( (9) \)

where G is:

\[
G = \begin{bmatrix} g_{1,1} & g_{1,2} & 0 & 0 & 0 & 0 & log(d_1^2) \\ g_{2,1} & g_{2,2} & 0 & 0 & 0 & 0 & log(d_2^2) \\ g_{3,1} & g_{3,2} & 0 & 0 & 0 & 0 & log(d_3^2) \end{bmatrix}
\]

\[
g_{1,1} = (x-x_{bi})/d_1^2 \\
g_{1,2} = (y-y_{bi})/d_1^2 \\
g_{2,1} = (x-x_{bj})/d_2^2 \\
g_{2,2} = (y-y_{bj})/d_2^2 \\
g_{3,1} = (x-x_{bi})/d_3^2 \\
g_{3,2} = (y-y_{bi})/d_3^2
\]

As it is seen in this case the path loss exponent is also brought in the state vector as state variable so that it can be estimated. \( x \) and \( y \) in the above equation is user coordination and \((x_{bi},y_{bi})\) is the \( i \)th BTS coordination.

B. Grid method and Particle filtering

In these methods their algorithm tries to estimate posterior probability density function (PDF) in the state space and finds its maximum (or calculates weighted average of the state space points) as the final solution, it is a kind of MAP technique that best estimates the solution. The starting point of these algorithms is based on the below equation:

\[
P( x | z) = \frac{P(z | x)P( x )}{P(z)} \propto P(z | x)P( x )
\]  \( (10) \)

where \( x \) and \( z \) are respectively state vector and measured parameter. \( P(x) \) is called prior information. So firstly the solution space (state space) should be approximated in some points, that is why these methods are classified as discrete [32, 33]. Based on how the initial points for state space created, Grid or Particle
method is emerged. If the space is uniformly divided it is called Grid, as its name implies, so there is a Grid like space to search for the optimal solution. Otherwise if the points in the solution space are created based on the prior probability distributed function particle filter is emerged. For example in Grid method if the solution space is 1D, then the mentioned PDF would be something like Fig.2, then mathematically it is possible to state the posterior probability in terms of sum of weighted and shifted delta functions.

The main idea of particle filtering generating a set of likely points for solutions in the state space and representing the probability distribution function only in those points by assigning and calculating a weight to each point and normalizing the weights. The set of points are generated by the prior information, it means, the more points are generated in the region with more probability for user to be there. In other words, some point in the solution space are generated according to prior PDF and then for each point, a corresponding weight is calculated and again the prior PDF is updated according to the resulting weights, repeating this process yields the solution. Calculating the weights is based on measurements. Mathematically suppose \( \{x_{0:i,k}, w_{k}^{i}\} : i = 1, ..., N_{s} \), to be the set of points and corresponding weights in the state space. The index i is for points and index k is for time. The weights are normalized so that \( \sum_{i} w_{k}^{i} = 1 \). The posterior probability is defined as follows [33, 34]:

\[
P(x_{0:k} | z_{1:k}) = \sum_{i=1}^{N_{s}} w_{k}^{i} \delta(x_{0:k} - x_{0:k}^{i})
\]

Which is approximated version of the real and continuous posterior probability. In the case that the weights cannot be calculated easily or generating the set of points according to the prior information is not possible, the importance sampling technique is used. For a particle filter to be efficient, the points which is generated in the state space should have as high weights value as possible to prevent degeneracy problem. One way of preventing the generation of low valued weights is resampling. Resampling means directly omitting the points with low valued weights and replacing them by high value weights point. Fig. 4 illustrates this process[35].

An example of particle filtering process in positioning with RSS parameter is developed in the simulation section with more details.

C. Metropolis Hastings Algorithm

This algorithm is basically used for generating random numbers with a target distribution and is in the class of Markov chain Monte Carlo method that uses acceptance rejection techniques to produce random numbers. It starts with generating a random number say \( x' \) from proposal distribution \( q(x|y) \) and suppose target distribution is \( \pi(x) \) and the initial state is known and equal to \( x=x_{0} \). If we define the ratio :

\[
r(x, x') = \frac{\pi(x') q(x'|x)}{\pi(x) q(x|x')}
\]

Then \( x=x' \) is the next state with the probability of \( r(x, x') \) otherwise the previous state (here \( x_{0} \)) is chosen for the next state. This algorithm converge to the final solution[35, 36].
But what is the application of this algorithm in positioning? See simulation section for details.

V. SIMULATIONS AND RESULTS

In this section some of popular methods have been simulated under the same and certain conditions to have a proper comparison. RSS measurement is used as observation parameter.

A. Network Model

To simulate the cellular network 3G system UMTS is chosen. A main cell and its 6 neighboring cells in the form of hexagonal shape is considered in MATLAB. See Fig. 4.

\[ X = \frac{8}{\sqrt{3} R} \] 0 \leq X \leq \frac{R}{2} \quad and \quad \sqrt{3}X < Y < \frac{\sqrt{3}R}{2} \]

\[ f_{x,y} = \begin{cases} 1 & \text{otherwise} \end{cases} \]

One easy way is to produce X coordinate uniformly in the interval 0 < X < R/2, then according to the conditions on Y, Y can be written in the form Y = \sqrt{3}X + V, but V can gets values such that the values of Y lies in its min/max constraints, i.e., V can have values between 0 and \( \frac{\sqrt{3}R}{2} - \sqrt{3}X \).

B. Kalman filtering simulation

According to (8) and (9), both position and path loss exponent are estimated in this simulation. At first a fixed user is simulated and in another simulation a path with discontinuity is assigned to the user and the corresponding RSS with path loss model (cost 231 Hata) are fed to the Kalman routine to estimate the position. Fig. 6, Fig. 7 and Fig. 8 show the process.

C. Genetic algorithm Simulation

The objective function of genetic algorithm is set according to (6) and (7). The other parameters are: Population size = 1000, Generation size= 1000; fitness
limit = 1 e -5. Tolerance function between successive generations = 1 e -10 . Fig.10 shows the process.

As stated before evolutionary algorithms is not suitable for real time processing because they are usually time-consuming. Here , the process of converging to the solution is so long if precise solutions are considered.

D. Particle filter simulation

For the simulation of particle filtering, 2 normal distribution with means equal to the main BTS coordination and standard deviation order of cell radius (R or R/2) is firstly chosen. Then 200 points for X and Y are generated and each is assigned a weight according to $P(x_{1:n} | p_i)$, where $p_i$ is the position of i\textsuperscript{th} point and n is the number of BTS that they are observed by the mobile station. note that this is likelihood function not posterior, but because the choosing point algorithm is used then the total process yields posterior distribution. Because the measurements of different BTS are independent then:

$$P(z_{1:n} | x) = \prod_{j=1}^{n} P(z_j | p_i)$$

and according to the (8) , $P(z_t | p_i)$ has log-normal distribution with mean $10 \alpha \ln(d)$ and variance equal to that of shadowing. After calculating the weights for each points, the weighted average is used for estimating the position for first iteration. For the second iteration, 200 points are drawn from the normal distribution with mean equal to the estimating position achieved from the first iteration. And the process goes for merely 2 or 3 iterations to reach the best point. Fig.10, Fig.11 and Fig.12 show the process, in this simulation the user location is in x= 100,y= 200 and R= 500. As it is shown the posterior PDF is going through the real position . For the second and third iteration, the same data size (200 points) are produced via random number generating algorithms extracting the normal random numbers but with the updated variance and mean from the data and their weights of the last iteration. (Mean and variance should be calculated regarding the weights of each point ).

E. Metropolis Hastings simulation

In the positioning target distribution is the posterior probability:

$$\pi(x) = P(p|z_{1:n})$$

where $p$ is the user position and $y_{t:n}$ is received power of n BTS’s. Proposal distribution $q(p’|p_{k-1})$ is defined as normal with mean $p$ and standard deviation which is order of cell radius (R or R/2):

$$q(p’|p) = N(p_{k-1}, \sigma^2)$$

E. Metropolis Hastings simulation

In the positioning target distribution is the posterior probability:

$$\pi(x) = P(p|z_{1:n})$$

where $p$ is the user position and $y_{t:n}$ is received power of n BTS’s. Proposal distribution $q(p’|p_{k-1})$ is defined as normal with mean $p$ and standard deviation which is order of cell radius (R or R/2):

$$q(p’|p) = N(p_{k-1}, \sigma^2)$$

Fig. 10 posterior approximation after iteration 1

Weight Function ( A posterior Probability )

Fig. 11 posterior approximation after iteration 2
So by this assumptions the corresponding ratio of MH\(^{17}\) algorithm is:

\[
r(p_{k-1}, p') = \frac{P\left((p' | y_n) q(p_{k-1} | p')\right)}{P\left((p_{k-1} | y_n) q(p' | p_{k-1})\right)}
\]  

(19)

where \(p'\) is the next generated point from proposal distribution, \(p_{k-1}\) is the last accepted point and \(p_0\) is the initial point. The outstanding property of normal distribution is manifested here. \(q(p_{k-1} | p')\) and \(q(p' | p_{k-1})\) has the mean \(p'\) and \(p_{k-1}\) respectively and they have the same values in the point \(p_{k-1}\) and \(p'\). so we have

\[
r(p_{k-1}, p') = \frac{P\left((p' | y_n)\right)}{P\left((p_{k-1} | y_n)\right)}
\]  

(20)

To calculate the numerator of the above ratio, Bayes rule is applied:

\[
P\left((p' | y_n)\right) = \frac{P\left(y_n | p'\right) P\left(p'\right)}{P\left(y_n\right)}
\]  

(21)

And for the denominator:

\[
P\left((p_{k-1} | y_n)\right) = \frac{P\left(y_n | p_{k-1}\right) P\left(p_{k-1}\right)}{P\left(y_n\right)}
\]  

(22)

From (8) and independent assumption of incoming signal from different BTS:

\[
P(y_n | p') = \prod_{j=1}^{n} P(y_j | p')
\]  

(23)

And \(P(y_j | p')\) is log-normal with mean \(10 \alpha \ln(d')\) and variance equal to that of shadowing. Fig.13 shows the MH process.

\(^{17}\) Metropolis Hastings
Even MH algorithm which seems to be less complex requires many iterations (more than 500,000 in the case of 4 variables) to converge to the solution as the number of state variables goes high while that of EKF is about 100 to reach even less error.

ACKNOWLEDGMENTS

This Study was supported by Islamic Azad University, Borujerd Branch, Iran. The authors would like to acknowledge staffs of university.

REFERENCES


Authors’ Profiles

Einollah Balarastaghi is an academic member of Islamic Azad University, Borujerd branch, Iran.

meh_balarastaghi@yahoo.com

Mohammad Reza Amini was born at 1981. He is an Iranian telecommunication researcher. He is an academic member of Islamic Azad University, Borujerd branch, Iran.

iman.amini@gmail.com

How to cite this paper: E. Balarastaghi, MR. Amini, A. Mirzavandi,”Simulation Based Comparison of Geo-Location Methods in Wireless Networks”, International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.2, pp.44-53, 2015. DOI: 10.5815/ijitcs.2015.02.06