AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

An Enhanced Localization Fusion Scheme for Mobile Sensor Networks Using Monte Carlo localization

[1]Dr. S. Mary Praveena, [2]Dr. Ila.Vennila, [3]Prof. S.B. Aneith Kumar, [4]Prof. M. Pravin

[1]*Associate Professor, Department of ECE, Sri Ramakrishna Institute of Technology, Coimbatore Ttamilnadu, India*
[2]*Associate Professor, Department of ECE, PSG College of Technology Coimbatore, Tamilnadu, India*
[3]Assistant Professor, *Department of ECE, Sri Ramakrishna Institute of Technology, Coimbatore Ttamilnadu, India*
[4]Assistant Professor, *Department of ECE, Sri Ramakrishna Institute of Technology, Coimbatore Ttamilnadu, India*

ARTICLE INFO

Article history:
Received 19 August 2014
Received in revised form
19 September 2014
Accepted 29 September 2014
Available online 3 November 2014

Keywords:
Mobile Sensor Networks, Localization and Sequential Monte Carlo methods.

ABSTRACT

Background: Localization in mobile sensor networks is more challenging than in static sensor networks because mobility increases the uncertainty of nodes positions. The localization algorithms used in the Mobile sensor networks (MSN)are mainly based on Sequential Monte Carlo (SMC) method. **Objective:** The existing SMC based localization algorithms commonly rely on increasing beacon density in order to improve localization accuracy and suffers from low sampling efficiency and also sampling in those algorithms are static and have high energy consumption. Those algorithms cannot able to localize sensor nodes in some circumstances. The main reason for that is in some time slots the sensor node cannot hear any beacon node. This results in localization failure. The Improved Monte Carlo Localization (IMCL) algorithm achieves high sampling efficiency, high localization accuracy even in the case when there is a low beacon density. **Result:** This can be achieved using bounding box and weight computation techniques. This algorithm also uses time series forecasting and dynamic sampling method for solving the problem of localization failure. Simulation results showed that the proposed method has a better performance in sparse networks in comparison with previous existing method.

© 2014 AENSI Publisher All rights reserved.

To Cite This Article: Dr. S. Mary Praveena, Dr. Ila.Vennila, Prof. S.B. Aneith Kumar, Prof. M. Pravin, An Enhanced Localization Scheme for Mobile Sensor Networks Using Monte Carlo localization. ***Aust. J. Basic & Appl. Sci.,*** *8(17): 426-434, 2014*

INTRODUCTION

Wireless sensor networks (WSNs) have been used in many fields, including environmental and habitat monitoring, precision agriculture, animal tracking, and disaster rescue. In many applications, it is essential for nodes to know their positions. For example, data should be labelled with the positions where they are collected to help the scientists perform corresponding analysis. Position information of nodes are also necessary in many network protocols, e.g., clustering and routing which depend on the geographical information of nodes. The procedure through which the nodes obtain their positions is called localization. In localization, the nodes in a sensor network can be categorized into two types: beacon nodes which are aware of their positions and sensor nodes which need to determine their positions using a localization algorithm. A straightforward method for localization in WSNs is to use existing localization techniques, e.g., attaching a Global Positioning System (GPS) receiver on every sensor node. However, as the scale of sensor networks becomes larger and larger, these methods become infeasible because of their high cost or inconvenience. In some recently emerging applications such as animal monitoring and tracking sensor nodes may move after deployment. These nodes form mobile sensor networks in contrast to traditional static sensor networks in which sensor nodes remain stationary after deployment. The motion of sensor nodes makes most existing localization algorithms designed for static sensor networks inapplicable to mobile sensor networks. There are some localization algorithms specially designed for mobile sensor networks, all of them are based on the Sequential Monte Carlo (SMC) method. This is because the SMC method provides simple simulation-based approaches in estimating the location. Previous SMC-based localization algorithms either suffer from low sampling efficiency or require high beacon density to achieve high localization accuracy. The major problem of most existing SMC-based localization algorithms is that they only rely on increasing beacon density to improve localization accuracy. However, beacon nodes are usually more expensive than sensor nodes. Because there are much more sensor nodes than beacon nodes in a sensor network, it will be very beneficial if sensor nodes can be used to improve the localization accuracy.

Corresponding Author: Dr. S. Mary Praveena, Associate Professor, Department of ECE, Sri Ramakrishna Institute of Technology, Coimbatore Ttamilnadu, India
E-mail: praveena_infant@yahoo.co.in

In this paper, we propose an efficient algorithm which addresses both aforementioned issues. The algorithm is based on the sequential Monte Carlo Localization (MCL) algorithm named as Improved MCL (WMCL). IMCL achieves high sampling efficiency and achieves high localization accuracy even when the beacon density is low using bounding box technique and weight computation methods. Despite the above technique haves good localization accuracy, sampling in these techniques are static and they have high energy consumption. Also the existing algorithms are not able to localize sensor nodes in some circumstances. The main reason is that in some time slots the node cannot hear any seed node. The Improved Monte Carlo Localization (IMCL) algorithm uses forecasting and dynamic sampling method for localization. This method has the ability of nodes localization in those conditions and it is an energy efficient method. The paper is organized as follows. The Section 2 deals with related work in localization of Wireless Sensor Networks. Section 3 deals with the proposed Improved Monte Carlo Localization (IMCL) algorithm. Section 4 is devoted to extensive performance analysis. Section 5 deals with the Conclusion and future directions.

MATERIAL AND METHODS

Extensive research has been done on localization for wireless networks. A general survey is done focusing only on localization techniques suitable for ad hoc sensor networks. The approaches taken to achieve localization in sensor networks differ in their assumptions about the network deployment and the hardware's capabilities.

Centralized localization techniques depend on sensor nodes transmitting data to a central location, where computation is performed to determine the location of each node. Doherty, Pister and Ghaoui developed a centralized technique using convex optimization to estimate positions based only on connectivity constraints given some nodes with known positions. MDS-MAP technique improves on these results by using a multidimensional scaling approach, but still requires centralized computation. Requiring central computation would be infeasible for mobile applications because of the high communication costs and inherent delay, hence we focus on distributed localization techniques.

Distributed localization methods do not require centralized computation, and rely on each node determining its location with only limited communication with nearby nodes. These methods can be classified as range-based and range-free. Range-based techniques use distance estimates or angle estimates in location calculations, while a range-free solution depends only on the contents of received messages. Range-based approaches have exploited time of arrival, received signal strength, time difference of arrival of two different signals (TDOA), and angle of arrival (AOA). Though they can reach fine resolution, either the required hardware is expensive (ultrasound device for TDOA, antenna arrays for AOA) or the results depend on other unrealistic assumptions about signal propagation (for example, the actual received signal strengths of radio signals can vary when the surrounding environment changes). Because of the hardware limitations of sensor devices, range-free localization algorithms are a cost effective alternative to more expensive range-based approaches.

Monte Carlo localization (MCL) method is developed for use in robotics localization for use in mobile sensor network applications. MCL is a particle filter combined with probabilistic models of robot perception and motion. It outperforms other proposed localization algorithms in both accuracy and computational efficiency. The key idea of MCL is to represent the posterior distribution of possible locations using a set of weighted samples. Each step is divided into a prediction phase and an update phase. In the prediction phase, the robot makes a movement and the uncertainty of its position increases. In the update phase, new measurements (such as observations of new landmarks) are incorporated to filter and update data. The process repeats and the robot continually updates its predicted location.

However, there are substantial differences between robot localization and node localization for sensor networks. While robot localization locates a robot in a predefined map, localization in sensor networks works in a free space or unmapped terrain. Second, a robot has relatively good control and probabilistic knowledge of its movement in a predefined map. A sensor node typically has little or no control of its mobility, and is unaware of its speed and direction. Third, a robot can obtain precise ranging information from landmarks, but a sensor node can only learn that it is within radio range. Finally, in robot localization, the individual measurements are integrated multiplicatively, assuming conditional independence between them, and the weights of samples need to be normalized after updating. In MCL, due to the constraints in computing and memory power, a filtering approach is adopted in which each measurement can be considered independently, and the weight of each sample is either 0 or 1.There are some other variants of MCL, for example, the dual and Mixture MCL, Multihop-based Monte Carlo Localization (MMCL), and Range-based MCL. The dual and Mixture MCL improves the localization accuracy of MCL by exchanging the probability functions used in the sampling step and in the filtering step. It incurs higher computational cost than MCL. MMCL and Range-based MCL use multihop sensor-beacon distances to improve the localization accuracy and to reduce the number of needed beacons. Compared with them, Our WMCL algorithm doesn't use multihop sensor-beacon distances so incurs much less communication cost and it uses weight computation methods to minimize localization error.The

previous existing methods have two major weak points which are not focused. The first problem is using of constant number of samples for localization. Second is that all nodes in all time slots cannot be localized.

Proposed imcl algorithm:

A. Introduction:

In IMCL algorithm, the network model is introduced and then the five main parts of IMCL are described as follows: Bounding-box construction, Dynamic sampling, Time series forecasting, Samples weights computing, maximum possible localization error computing.

B. Building the Bounding Box:

There are two areas involved in bounding box of IMCL: the candidate samples area and the valid samples area. The candidate samples area is used to draw new candidate samples and the valid samples area is used to filter out invalid samples. When the candidate samples area is large and the valid samples area is small, candidate samples drawn in the sampling step have high probability to be filtered out in the filtering step.figure1 shows the construction of bounding box in IMCL algorithm.

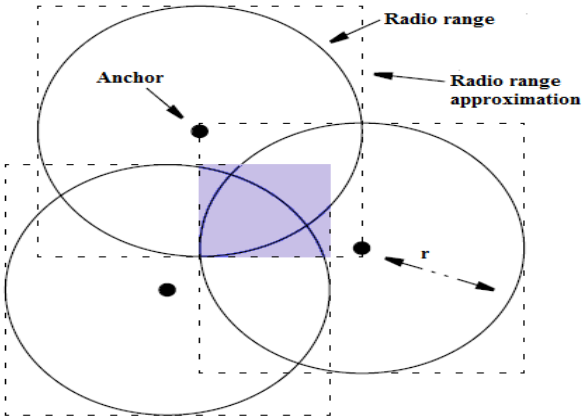

Fig. 1: Building the Bounding-box.

In IMCL, the possible locations of a sensor node after move lie in a disk with radiusV_{max} . So the size of the candidate samples area will increase when V_{max} increases. On the other hand, when sd increases, the size of the valid samples areas will decrease. Denote by V_t the total number of candidate samples drawn in the sampling step in time unit t and define the sampling efficiency in t as,

$$e_t = \frac{|L_t|}{Vt} \quad (1)$$

Then in WMCL the sampling efficiency will decrease when V_{max} or Sd increases, which will cause high computational cost accordingly. Two-Hop beacon neighbours are also used to reduce the size of the bounding-box by replacing r with 2r. The candidate samples are chosen from the bounding box.

B. Dynamic sampling method:

As previously mentioned, instead of taking a fixed number of samples like 50 for localization, we can determine this number dynamically based on the size of the sampling area. It is clear that for a large anchor box, a large number of samples are needed to estimate nodes location accurately. While in the case of small anchor box, we will focus on a small area. For a small area, small number of samples is needed to accurately estimate nodes position.

If created anchor box have the coordinates of (X_{min}, Y_{min}) and (X_{max}, Y_{max}), so area size of this box is determined and we will specify the number of samples based on this area size. Noting to the standard number of samples that is 50, this number will be used for an anchor box with maximum area size. An anchor box is maximized when the node hears only a one-hop anchor node. In this case the box size will be equal to a square of size $2V_{max}$. So we will consider 50 samples for this box and use equation (5) for other box sizes.

$$\text{Sample Number}=50*((x_{max}-x_{min})(y_{max}-y_{min}))/_{4V_{max}^2} \quad (2)$$

When the anchor box has the maximum area, numerator and denominator of the fraction in the equation (2) will be equal and so the number of samples will be equal to 50.

For the cases when the anchor box size is more than $4V_{max}^2$, we consider the number of samples equal to 50. Example of such cases is when the sensor node hears only one two-hop anchor node.

C. Linear prediction using time series:

Linear prediction method is a powerful technique for predicting time series in a time-varying environment. This method is expressed in equation(6) and is a recursive method

$$y(t + T) = a_1y(t)+a_1y(t-T)+ ... +a_my(t-(m-I)T) \quad (3)$$

Estimated value at time t as a linear function of previous values in the times "t-T, t-2T. .. t-mT" has been produced is obtained. In equation(3) a1, a2, ..., am are the linear prediction coefficients, 'm' is the model degree, 'T' is the sampling time, y(t+T) is the next observation estimation and y(t), y(t-T), ..., y(t-mT) are the present and past observations. The prediction error which is the difference between the predicted and the real locations (Equation (4))must be minimized.

$$\text{Error}(\%)= \{\frac{predictedlocation \ -Reallocation}{Reallocation}\}* 100\% \quad (4)$$

In order to estimate the coefficients of linear prediction model we use the least squares error method and rewrite equation (3) with considering modelling error in equation (4):

$$y(t) = a_1y(t)+a_1y(t-T)+ ... +a_my(t-(m-I)T) +e (t) \quad (5)$$

The error e(t) is generated because of not adopting the linear prediction model to the real value. So to find the coefficients, a1, a2, …… a_m in equation(5), we use the sum least squares error and set of linear functions presented in equation(6)

$$\begin{bmatrix} y(t) \\ y(t-T) \\ M \\ y(t-mT) \end{bmatrix} = \begin{bmatrix} y(t) & y(t-T) & \wedge & y(t-mT) \\ y(t-2T) & & \wedge & y(t-(m+1)T) \\ M & & M & \\ y(t-(k+1)T) & & \wedge & y(t-(m+k)T) \end{bmatrix} * \begin{bmatrix} a1 \\ a2 \\ M \\ am \end{bmatrix} + \begin{bmatrix} e(t) \\ e(t-T) \\ M \\ e(t-kT) \end{bmatrix} \quad (6)$$

$$Y=\phi+A+E \quad (7)$$

Elements in the matrix A are the coefficients which can be found by least squares error method in equation (8):

$$A=(\phi^T\phi)^{-1}\phi^TY \quad (8)$$

In equation (8), ϕ^T is the transpose of the matrix ϕ and $(\phi^T\phi)^{-1}$ is the inverse of matrix. After obtaining the coefficients a1, a2 ... am, the nodes location in the next time slot predicted using equation (3). If the node do not localized using WMCL algorithm with dynamic sampling, we will use this predicted location instead. Then the weights are computed for predicted samples.

D. Weighting the Samples:

After a sample candidate is chosen, its weight is computed using 1-hop and 2-hop (anchor and common) neighbour nodes. Figure2 shows the phenomenon of weight computation.

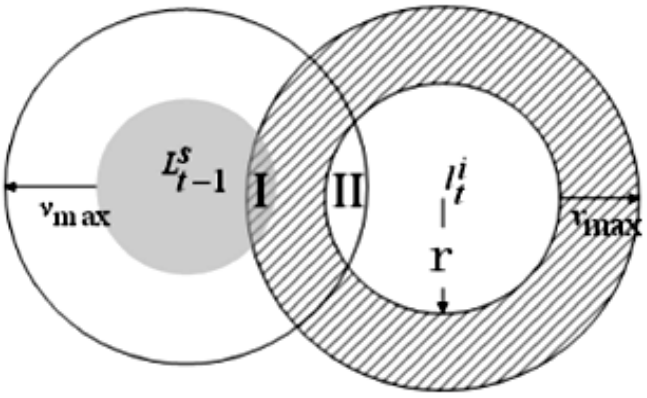

Fig. 2: Weight Computation Method.

In WMCL, $O_t = S \cup T \cup US$. So the weight of a candidate sample is computed as

$$w_t = p(O_t | l_t) = \prod_{s\in S,T} p(s \mid l_t) \prod_{s\in US} p(s \mid l_t) \quad (12)$$

When $s\in S$ or $s\in T$, $P(s \mid l_t)$ can be easily computed

If $s\in S$, then

$$P(s \mid l_t) = [d(l_t,s) \le r] \quad (13)$$

If $s\in T$, then

$$P(s \mid l_t) = [r < d(l_t,s) \le 2r] \quad (14)$$

The weights of samples are computed. The samples with zero weights are rejected and samples with high weights are taken for error computation.

E. Error Computation:

After obtaining N valid samples, a sensor node computes the weighted average of these samples as its position estimation. Using the position estimation and the bounding- box, a sensor node can compute its ER_x and ER_y, as illustrated in Figure3.

A more riskily method is to use the smallest rectangle enclosing all of p's valid samples to compute ER_x and ER_y. This method can improve localization accuracy a lot. However, when using this method the procedure of constructing the bounding-box should be carefully manipulated. In this case the inequality p causes some inconsistence in the computation.

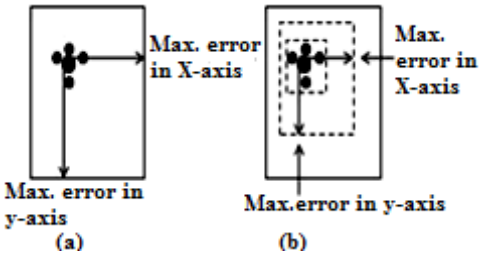

Fig. 3: Computing maximum localization error.

For example, it is possible that x_{min} is larger than x_{max} and consequently the bounding-box cannot be built. After p gets (x_e,y_e) and ER_x, ER_y, it broadcasts them to its neighbors. Its neighbors will use this information to compute their position estimation in the next time unit. Algorithm of the proposed method has been presented in figure (4).

```
if ( ObservationSet_t ≠ ϕ ) //The node see one or several anchor nodes
L_t = ϕ
Compute N // According to Equation6
While (size (L_t) < N)
Run WMCL Algorithm
Location Estimation[i] = Σ_{i=1}^{N} l_t^i / N
If trend line exist
Update trend line in TSF
else
Return (-1)
else if (Number of Location Estimation[i]>=4)
Create trend line with Location Estimation[i]
Calculate < a1,a2, ... ,am>←est_ Coefficient()
// estimate a1,a2, ... ,am based on the Modeling Window
Calculate Location Estimation[i]
```

Fig. 4: IMCL Algorithm

RESULTS AND DISCUSSIONS

The simulation is carried out in NS-2 simulator under Linux platform with simulation area of 1000 x 1000m and 150 mobile nodes.The node is randomly placed from there onwards the node mobility occurs in random direction. The simulation time is 100 seconds. The simulation parameters are given in table 1.

Table 1: Simulation Parameters in NS-2

PARAMETER	RANGE
Simulation Area	1000m x 1000m
Simulation time	100 Sec
No. of Sensor Nodes	150
No. of Beacon Nodes	8
Communication Range(Broadcasting)	250m
Node Mobility Max Speed	0-20 metre/sec
Propagation model	Free space propagation
MAC type	MAC 802.11
Routing protocol	AODV

The performance of the proposed IMCL algorithm is compared with the existing WMCL algorithm using various performance metrics and the results are obtained as follows:

A. Sampling efficiency:

The sampling efficiency is a very import metric in SMC-based localization algorithms because higher sampling means less candidate samples generation and consequently less computational cost.

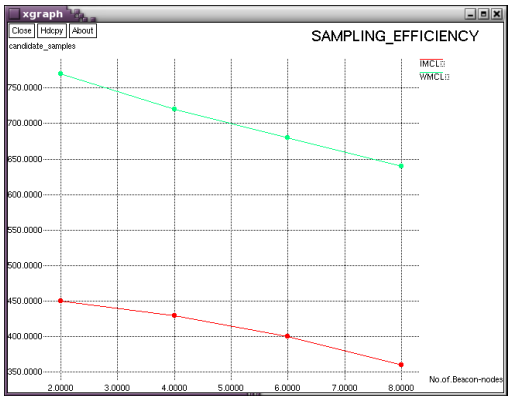

Fig. 5: Samples Vs Beacon nodes.

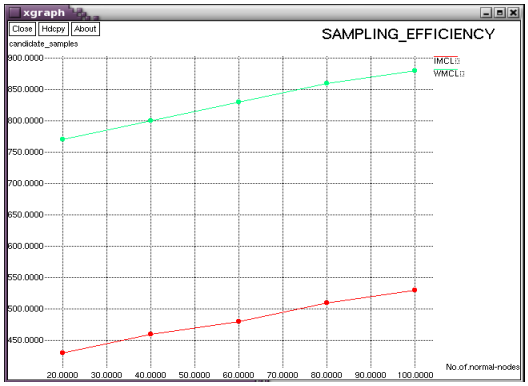

Fig. 6: Samples Vs Sensor nodes.

Figure (5) & (6) shows the number of candidate samples obtained for localization in the WMCL and IMCL algorithms with varying number of beacon and sensor nodes. The number of samples in WMCL algorithm is fixed and each unknown node during each time slot uses 50 samples to do localization. But in the IMCL algorithm samples number is determined dynamically. Simulation results show that with using IMCL algorithm, less candidate samples are chosen thereby obtaining high sampling efficiency and each unknown node uses fewer samples than other methods.

B. Sampling Attempts:

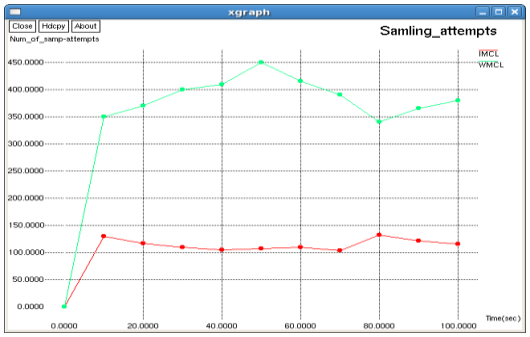

Fig. 6: Sampling attempts

Most of computational energy consumption for these algorithms is related to the number of used samples and the number of sampling attempts to produce acceptable samples. Also the response time depends on the

number of sampling attempts for production of required samples. Figure (6) shows simulation results for the number of sampling attempts to produce enough valid samples.

C. Dynamic Sampling:

The Dynamic sampling is performed in mobile nodes. Figure (7) shows the number of valid samples obtained for localization in the WMCL and IMCL algorithms. The number of samples in WMCL algorithm is fixed and each unknown node during each time slot uses 50 samples to do localization. But in the IMCL algorithm samples number is determined dynamically. Simulation results show that with using IMCL algorithm, each unknown node uses fewer samples than WMCL method.

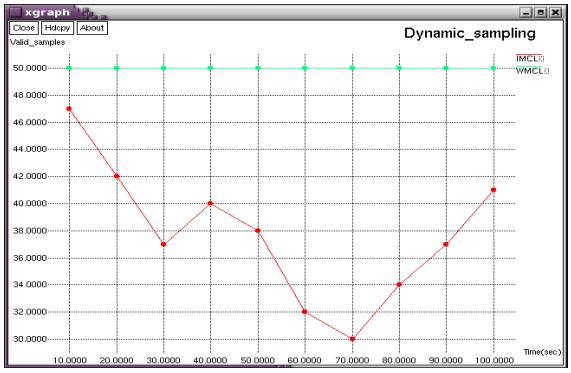

Fig. 7: Dynamic sampling.

D. Localization Accuracy:

Localization accuracy is the most important metric in evaluating localization algorithms. The localization accuracy is determined from the estimated value of localization error. The localization error is noted for different time periods. The localization error is also determined by varying number of beacon and sensor nodes.

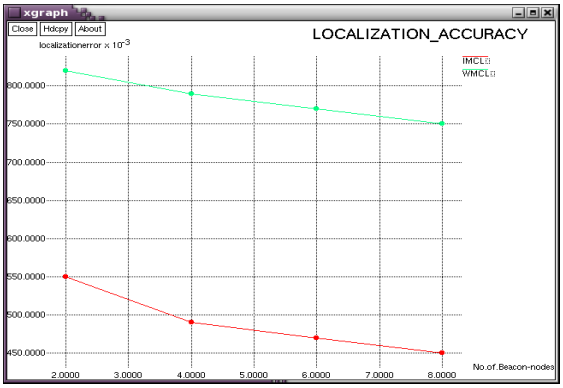

Fig. 8: Localization errorVs Beacon nodes.

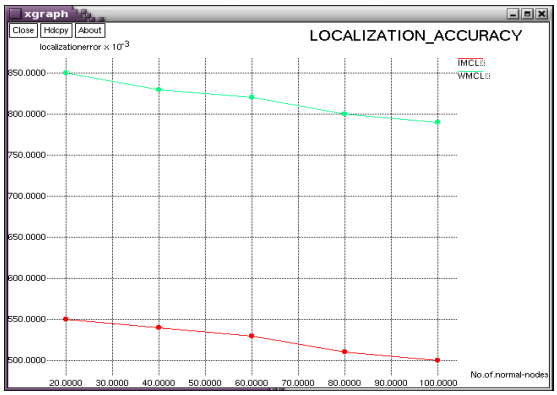

Fig. 9: Localization error Vs sensor nodes.

Figure(7) & (8) shows the localization error for varying number of beacon and sensor nodes. The graph shows that the localization error in IMCL algorithm is reduced compared with WMCL algorithm and thereby having high localization accuracy.

E. Energy backlog:

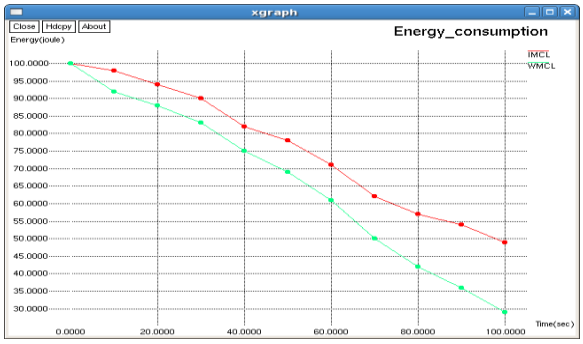

Fig.10: Energy backlog.

The dynamic sampling method and TSF method used in IMCL algorithm will reduce the energy consumed in localization process by minimizing the number of sampling operation. Fig 10shows the amount of energy consumed in IMCL algorithm is very much reduced than the WMCL algorithm.

Conclusion:

The Improved Monte Carlo Localization (IMCL) algorithm achieved high sampling efficiency, high localization accuracy even in the case when there is a low beacon density using the bounding box and weight computation methods. The localization accuracy is improved by using the estimated position information of sensor nodes. The proposed IMCL algorithm used dynamic sampling based on the size of sampling area to estimate the sensor nodes position. Also the proposed algorithm uses TSF method to predict sensors position when the sensor nodes do not hear any anchor nodes. The proposed algorithm is suitable for mobile sensor networks with low anchor node density. This algorithm has less implementation costs in comparison with previous method. Simulation results showed that the proposed algorithm provides better performance than the similar method in the sparse sensor networks.

REFERENCES

Baggio and Langendoen, 2006. “Monte-Carlo Localization for Mobile Wireless Sensor Networks”, Proc. Conf. Mobile Ad-Hoc and Sensor Networks (MSN ’06), pp: 317-328.

Boukerche, A., H. Oliveira, E. Nakamura and A. Loureiro.,2008. "Secure localization algorithms for wireless sensor networks," *IEEE Commun. Mag.*, 46(4): 96-101.

Datta, S., C. Klinowski, M. Rudafshani and M. Khaleque, 2006. “Distributed Localization in Static and Mobile Sensor Networks,”Proc. IEEE Int’l Conf. Wireless and Mobile Computing, pp: 69-76.

Hu, L. and D. Evans, 2004. “Localization for Mobile Sensor Networks,”Proc. ACM MobiCom, pp: 45-57.

Hill, J., R. Szewczyk, A. Woo, S. Hollar, D.E. Culler and K.S.J. Pister, 2000. "System architecture directions for networked sensors," in Proceedings ofASPLOS, pp: 93-104.

Karp, B. and T. Kung, 2000.“GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” Proc. ACM MobiCom, pp: 243-254.

Lin, T.H., P. Huang, H.H. Chu and C.W. You, 2009. "Energy-efficient boundary detection for rf-based localization systems," *IEEE Trans. Mobile Comput.*, 8(I): 29-40.

Lim, H. and J. Hou, 2005. “Localization for Anisotropic Sensor Networks ”Proc. IEEE INFOCOM, pp: 138-149.

Soltaninasab, B. and J. Amiri, 2010. “Improving Monte Carlo Localization Algorithm Using Time Series Forecasting Method and Dynamic Sampling in Mobile WSNs,” Proc. Conf. Communication Systems, Networks and Applications, pp: 389-396.

Tilak, S.V. N.B. Kolar, Abu-Ghazaleh and K.D. Kang, 2004. "Dynamic localization protocols for mobile sensor networks ," in Proceedings of MASS, pp: 56-67.

Shigeng, Z. J. Cao, C. Lijun and C. Daoxu, 2010. “Accurate and Energy-Efficient Range-Free Localization for Mobile Sensor Networks”, IEEE transactions on mobile computing, 9: 897-909.

Shigeng, Z.J. Cao, C. Lijun and C. Daoxu, 2008. “Location Nodes in Mobile Sensor Networks More Accurately and Faster,” Proc. Ann.IEEE Conf. Sensor and Ad Hoc Comm. and Networks, pp: 76-87.

Wang, W. and Q. Zhu, 2007. "Varying the Sample Number for Monte Carlo Localization in Mobile Sensor Networks," Proc. IEEE Int'l Multi-Symp. Computer and Computational Sciences, pp: 490-495.

Zeng, Y., J. Cao, S. Zhang, S. Guo and L. Xie, 2009. "Pollution attack: A new attack against localization in wireless sensor networks ," in Proceedings of WCNC, pp: 76-85.

Zeng, Y., S. Zhang, S. Guo and X. Li, 2007, "Secure hop-count based localizationin wireless sensor networks," in Proceedings of CIS, pp: 89-94.

Zeng, Y. and J. Cao, 2009. "SecMCL: A Secure Monte Carlo Localization Algorithm for Mobile Sensor Networks ", Proceedingsof Conference. Mobile Ad-Hoc and Sensor Networks, pp: 56-62.