2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)
Mexico City, Mexico. November 11-13, 2020

Self-localization of Sensor Node Using Monte Carlo Method Considering Shadowing

Timoteo Cayetano-Antonio
UMI-LAFMIA
CINVESTAV-IPN
Mexico City, Mexico
tcayetano@cinvestav.mx

M. Mauricio-Lara
UMI-LAFMIA
CINVESTAV-IPN
Mexico City, Mexico
mlara@cinvestav.mx

Aldo G. Orozco-Lugo
UMI-LAFMIA
CINVESTAV-IPN
Mexico City, Mexico
aorozco@cinvestav.mx

***Abstract*—The sequential Monte Carlo method for localization in Mobile Wireless Sensor Networks has been extensively studied using free space as the communication channel model. Besides, numerous and limited-hardware sensors have been assumed when proposing different algorithms based on this method. However, there is some lack of information on its performance using a more realistic shadowing channel model. In this paper, a proposal of sensor node localization using Monte Carlo method is presented. We consider shadowing and more powerful sensors to study how this kind of method behaves.**

***Index Terms*—Mobile Wireless Sensor Node, Localization, Positioning, Monte Carlo Method, Particle Filter, Shadowing, RSSI.**

I. Introduction

Wireless Sensor Networks (WSN) are a large set of low-cost sensor nodes that measure or collect certain information in its environment and share its data wirelessly [1]–[4]. WSN have specific characteristics which makes them suitable for environmental monitoring, home automation, medical care and disaster prevention, among others [5], [6]. These characteristics, which make them different from other ad-hoc networks, represent new and interesting challenges in the management of this kind of networks.

One of these challenges is the localization or positioning of a sensor in a WSN. In WSN applications, localization plays a key role since, for each node, it is mandatory to know its position in order to get a correct and meaningful interpretation of the sensed value. For this reason, considerable amount of research on localization for static WSN have been made. For mobile WSN, as they represent a more difficult challenge to face, the amount of research is reduced.

In the past decade, the Monte Carlo Method has been studied to deal with the localization problem in mobile WSN. Since the first publication [7], a significant number of works has been presented based on this method [8]. Although there are interesting works that present algorithms with good performance, most of them take idealized assumptions that can lead to misleading results about the real performance of the method on real environments. The strongest assumption is that nodes work on a free space communication environment. Besides, these works assume that nodes are extremely limited in hardware; which is not the case nowadays [9]. Moreover, in Monte Carlo methods, accuracy of localization strongly depends on the number of deployed sensor nodes [7]; however, in practice, this number is not as high as it is assumed.

Because of these reasons, in this paper, we propose a new algorithm based on the Monte Carlo method for node localization in WSN in a more realistic communication channel model with shadowing. Besides, we assume that the sensor nodes can estimate their velocity and that they can only use their own measurements without the help of other sensor nodes, as is the case for the Monte Carlo algorithms presented in previous works.

This paper is organized as follows. In Section II the proposed algorithm is presented alongside the description of the localization problem in mobile WSN. In Section III, simulations are carried out; and, finally, in Section IV, conclusions of the work are given.

II. The algorithm

The localization problem scenario in this work can be stated as follows. A WSN is considered to have two type of nodes: the nodes of the first type, called seeds or anchors, know its position all along; the nodes of the second type, called simply sensor or sometimes blind nodes, need to estimate its location based on the position that seeds broadcast periodically. It is assumed that all nodes can move randomly within a limited region. Without loss of generality, we concentrate on a particular sensor node in the network which moves according to a random waypoint mobility model. Moreover, this node has the ability to measure its velocity according to the following model: $v_m = v_t + z_{\sigma_v}$, where v_m is the measured velocity, v_t is the true velocity and z_{σ_v} is the noise modeled as zero-mean gaussian random variable with variance σ_v^2.

Henceforth, the notation used is as follows. $x_k \in \Re^2$ is the 2D real position of node at instant k; whereas $\hat{x}_k \in \Re^2$ is the estimated position. o_k is the measurement or observation received by the node at instant k, i.e. the position that a seed node has broadcast at instant k.

A. The observational model

In this work, the log-normal shadowing communication channel model is used [10]. In this model the path loss is expressed as

$$PL = PL_0 + 10 n_p \log \frac{d}{d_0} + X_\sigma \tag{1}$$

where PL is the power loss of a transmitting antenna to a distance d; n_p is the decay exponent and PL_0 is the power of reference calculated at a distance d_0 in free space. In addition, X_σ is a zero-mean gaussian distribution in dB with a standard deviation σ_X. From Equation (1), the distance can be expressed in terms of PL as

$$D = 10 \log \frac{d}{d_0} = (PL - PL_0 - X_\sigma)/n_p \qquad (2)$$

where D is the log-distance between the transmiting and the receiving antennae. It is clear that D is also gaussian with the same variance as PL, σ_X. As such, it follows the behaviour of a gaussian distribution in the sense that 99% of its values lie within three standard deviations of the mean [11]. This can be stated with the following probability, Pr:

$$Pr(D - 3\sigma \leq \bar{D} \leq D + 3\sigma) \approx 0.99 \qquad (3)$$

B. The Monte Carlo Method

The localization problem can be modeled as a nonlinear filter by means of this stochastic model [12]:

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}, v_{k-1}) \\ o_k &= g_k(x_k, w_k) \end{aligned}$$

which indicates that the node position x_k is a function of its last position x_{k-1} and a process noise, v_{k-1}; whereas the observation is a function of the current position, x_k and a process noise w_k.

The estimate of x_k is sought based on the sequence of all available measurements from instant 1 to k, $o_{1:k}$. From the Bayesian point of view, the problem is to recursively quantify some belief in the state x_k, given the sequence of observations $o_{1:k}$. Thus, it is required to construct the a posteriori probability density function (pdf), $p(x_k|o_{1:k})$. With the knowledge of this pdf, it is possible to estimate the state $\hat{x}_k$ with some criterion [12].

The main idea of Monte Carlo method is to approximate the posterior distribution $p(x_k|o_{1:k})$ by a set of random samples, x_k^i with associated weights, w_k^i, and to express it as

$$p(x_k|o_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i) \qquad (4)$$

where $\delta(\cdot)$ is the Kronecker function and N_s is the number of samples used to approximate the distribution.

The weights are expressed in terms of the importance probability function; and this probability function is chosen as the transitional probability state, $p(x_k|x_{k-1})$ [13]. Consequently,

$$w_k^{*i} = w_{k-1}^i p(o_k|x_k).$$

The weights are then normalized, in suchs a way that, $\sum_{i=1}^{Ns} w_k^i = 1$. Thus,

$$w_k^i = \frac{w_k^{*i}}{\sum_{j=1}^{N_S} w_k^{*j}} \qquad (5)$$

Therefore, the Monte Carlo method consists of an approximation of the pdf $p(x_k|o_{1:k})$ by means of a set of samples $\{x_k^i\}_{i=1}^{N_S}$ as Equation (4) shows. These samples are obtained from the importance sampling distribution function, $p(x_k|x_{k-1})$. Finally, all the samples are weighted by the observational pdf and normalized as Equation (5) indicates.

The above method will exhibit the unavoidable degeneracy problem [12]; this is that, after a few iterations, only few samples will have high weights. In order to cope with this problem, a resampling process is necessary. The resampling is done from the obtained samples. Resampling is necessary when $N_{eff} < N_{threshold}$ [13]; where

$$N_{eff}(k) = \frac{1}{\sum_{i=1}^{N_S} (w_k^i)^2} \qquad (6)$$

C. The proposed Monte Carlo algorithm of self-localization

This algorithm scheme is based on the particle filter previously explained. However, some modification has been done in order to improve the performance of the localization problem. In this scheme, only one sensor node is considered; hence it will self-localize using the observations received by the seed nodes. These observations are the RSSI expressed in terms of the power given in Equation (1).

It was explained that the samples are drawn from the importance distribution function $p(x_k|x_{k-1})$. With the measurement of the velocity, v_m, this importance function can be defined as,

$$p(x_k|x_{k-1}) = \begin{cases} \frac{1}{\pi v_m^2} & \text{if} \quad d(x_k, x_{k-1}) \leq v_m \\ 0 & \text{if} \quad d(x_k, x_{k-1}) > v_m \end{cases} \qquad (7)$$

where $d(\cdot,\cdot)$ represents the euclidean distance between two 2D points. It is necessary to clarify that v_m is put in units of distance and represents the distance that a node can move between two instants of localization. Equation (7) states that the sample at instant k is drawn from a circular region which center is x_{k-1} and radius is v_m.

As it can be seen, the observations are not used in the definition of the importance function. Thus, the region where samples are drawn depends exclusively on v_m. Consequently, if the node is moving with great velocity, this region will be large; impacting in the accuracy of the localization. Taking this into account, our algorithm uses not only $p(x_k|x_{k-1})$ as the Monte Carlo method dictates but also the observations received from seeds in order to reduce the area where samples are drawn, and therefore the localization error. This observation is used as explained below.

Based on Subsection II-A; if a blind node is able to receive a signal from a seed, the log-distance between these two nodes is most probably found in the interval given in (3). Figure 1 shows a node which has received the signal from 3 seeds. A rectangle has been used to delimitate the region where this node can be likely found. Therefore, in our method the prediction zone is not only defined by the circumference stated in (7) but also by the observations delimited by the rectangular regions shown in Figure 1.

Now, the assignment of weights of the samples drawn in the prediction zone previously defined are done in the following

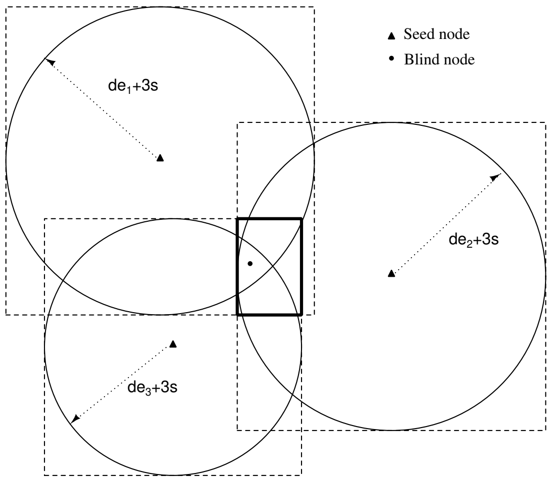

Fig. 1. Prediction zone using only measurements received by 3 seed nodes. For each seed node; the sensor estimates the log-distance de_j, according to the measurement received, and adds $3\sigma_X$ because of (3). $s = \sigma_X$.

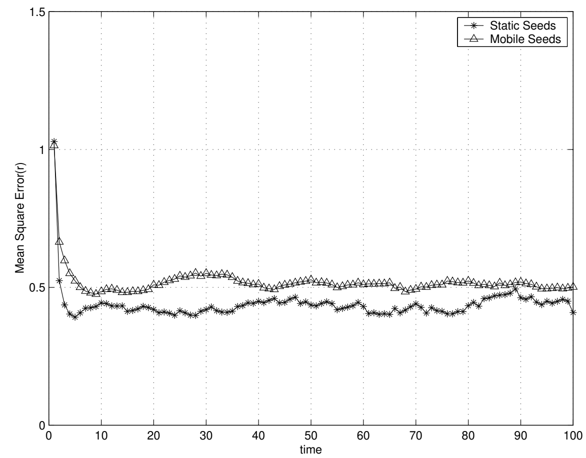

Fig. 2. Transient study

manner. Assuming that the sensor receives the signal of J seeds, the non-normalized weight of sample i is

$$w_k^{*i} = w_{k-1}^i \Pi_{j=1}^J p(o_k|x_k) \tag{8}$$

as the shadowing is considered uncorrelated. $p(o_k|x_k)$ is established in the observational model expressed in Equation (1). The normalized weight is obtained according to (5). If resampling is necessary; our algorithm uses the same criterion expressed in [13] and Equation (6).

Finally, the pseudocode is shown in Algorithm 1.

Algorithm 1 MCL Self-localization

Input: σ_X, N_S, $N_{threshold}$,v_{max}, v_m
Output: Node position

Initialization: The first sample set $L_0 = \{x_0^i\}_{i=1}^{N_S}$ is obtained from $p(x_0)$; and $\{w_0^i = 1/N_S\}_{i=1}^{N_S}$.
Localization at instant $k \neq 0$**:**$L_k = \{\}$
while size$(L_k) < N_S$ **do**
  for $i = 1$ **to** $|L_{k-1}|$ **do**
    Define the prediction region as the intersection of the approximation of rectangles as shown in Figure 1.
    Reduce the prediction region with intersection of the circle centered in x_{k-1}^i and radius v_m.
    Predict sample x_k^i inside prediction region.
    Assign sample according to Equation (8).
    $L_k = L_k \cup \{x_k^i\}$
  end for
end while
Normalize weights according to (5)
Calculate N_{eff} according to (6)
if $N_{eff} < N_{threshold}$ **then**
  Resampling
end if
Estimation:
Node location = $\frac{\sum_{i=1}^{N_S} x_k^i}{N_S}$

III. SIMULATIONS

We performed a set of simulations under the following considerations. Unless otherwise stated, there are 10 nodes: just one of them is the blind node under study and the rest are seed nodes. Besides, both a sensor or a seed node has a nominal coverage radio $r = 30$ meters. The nodes are moving inside a square region of 100×100 m^2 following the random waypoint mobility model. The seed nodes, when they are able to move, follow the same model. Furthermore, all nodes can move with randomly chosen velocity in the range $[v_{min}, v_{max}]$; where $v_{min} = 0.1$ and, if not specified, $v_{max} = 5$. On the other hand, the sensor node is able to measure its velocity following a gaussian error measurement model with $\sigma_v = 0.1$. In addition, we assume a communication channel with shadowing. If it is not indicated, the shadowing is characterized with $\sigma_X = 4$ and $n_p = 2.3$. Finally, it is worth mentioning that the number of samples used to approximate $p(x_k|o_{1:k})$ is 50.

Simulations are carried out employing different parameter values. In order to evaluate the performance of the algorithm we use the error defined as $Error(k) = ||x_k - \hat{x}_k||/r$ which gives the localization error at instant k. This error is averaged over 100 independent realizations.

Figure 2 shows the performance of the transient response for the algorithm for both static and mobile seeds. These results indicate that after a number of localization instants, the error reaches the steady state in the mean. Then, in the following simulations the error in each plot are the error in the steady state.

In the second simulation carried out, the maximum velocity was varied. The results are shown in Figure 3. In this study, and in the followings, we add the performance of the multilateration localization method in order to obtain a comparison with the proposed algorithm. The results show that maximum velocity does not affect much the performance of multilateration. However, in terms of error, it remains over the proposed Monte Carlo algorithm for both static and mobile seeds.

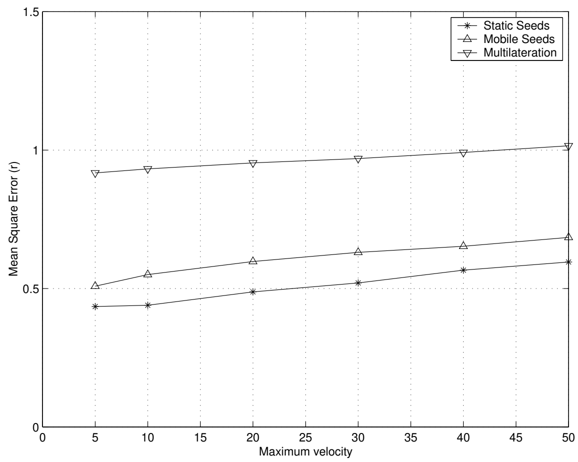

Fig. 3. Error for different maximum velocity.

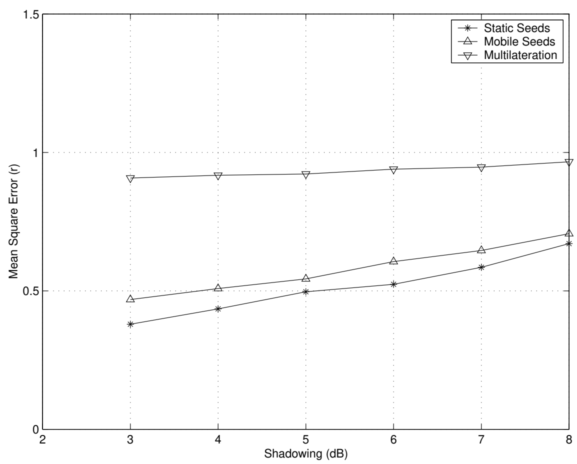

Fig. 4. Error for different Shadowing.

In Figure 4, results of varying shadowing variance, σ_X^2, are shown. It is remarkable to note how this parameter affects the performance of the localization algorithms when the other parameters remain unchanged. However, the proposed Monte Carlo algorithm cope better with this than its multilateration counterpart.

Finally, we evaluate the performance of the proposed algorithm for different number of seeds. The results are shown in Figure 5. As expected, the larger the number of seeds, the greater the accuracy of the localization algorithms.

IV. Conclusions

In this paper, a localization algorithm for WSN based on the Monte Carlo method is proposed. In our work we have used a communication channel which considers uncorrelated shadowing which is a more realistic model than the free-space model considered in previous works. Also, contrary to other works, the sensor takes into account just the observations of one-hop seeds. Furthermore, we considered a sensor which is able to measure its velocity; i.e. a more powerful sensor according to the current state of technology.

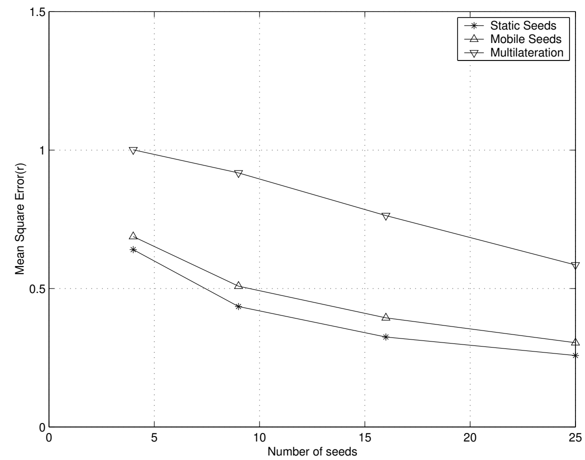

Fig. 5. Error for different number of seeds.

Simulations show that our proposal presents a better performance when the seed nodes are static than when they are moving. This is noticeable since our algorithm does not take into account this fact. The most important factor that affects the performance of the proposal is the unpredictability given by the shadowing. In this paper, we have used the uncorrelated shadowing. Work is on their way considering an algorithm that takes into account correlated shadowing.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE communications magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] R. R. Selmic, V. V. Phoha, and A. Serwadda, *Wireless Sensor Networks*. Springer, 2016.

[3] M. S. Obaidat and S. Misra, *Principles of wireless sensor networks*. Cambridge University Press, 2014.

[4] I. M. El Emary and S. Ramakrishnan, *Wireless Sensor Networks*. CRC Press, 2013.

[5] M. Ilyas and I. Mahgoub, *Smart Dust: Sensor network applications, architecture and design*. CRC press, 2006.

[6] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware design experiences in zebranet," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 227–238.

[7] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 45–57.

[8] A. Znaid, M. Ammar, M. Idris, Y. Idna, A. W. Abdul Wahab, L. Khamis Qabajeh, and O. Adil Mahdi, "Sequential monte carlo localization methods in mobile wireless sensor networks: a review," *Journal of Sensors*, vol. 2017, 2017.

[9] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, and N. M. Khan, "A critical analysis of research potential, challenges, and future directives in industrial wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 39–95, 2017.

[10] T. S. Rappaport, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.

[11] E. Grafarend and J. Awange, *Linear and Nonlinear Models: Fixed Effects, Random Effects, and Total Least Squares*. Springer, 2012.

[12] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house, 2003.

[13] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.