Advanced Research in Electrical and Electronic Engineering
Print ISSN: 2349-5804; Online ISSN: 2349-5812 Volume 2, Number 8 April-June (2015) pp. 84-88

http://www.krishisanskriti.org/areee.html

Kalman Filter and Monte Carlo Localization in Wireless Sensor Network

Shweta Singh[1], Ravi Shakya[2] and Yaduvir Singh[3]

[1,2,3]*Department of Computer Science, Ideal Institute of Technology, Ghaziabad*
E-mail: [1]shweta.svu@gmail.com, [2]ravisshakya@gmail.com, [3]yaduyash@gmail.com

Abstract—*Location information of sensor nodes is the key information which is used for routing, target tracking, deployment, rescue or coverage etc… Since wireless sensor network consists of a large number of low cost sensor nodes which are deployed over a wide area and used for environmental monitoring, battle field surveillance, health surveillance; it is required to know the position of sensor nodes. For researcher this localization estimation is the significant challenge. GPS (Global positioning system) is one of the widely accessible and accurate techniques used for location estimation of sensor nodes. The drawback of GPS is its high cost and energy consumption. To reduce the energy consumption and cost some beacon nodes which contain the GPS modules are deployed; other nodes determine their location using localization techniques. In 1960 R.E. Kalman gave a recursive approach to the discrete-data linear filtering problem. To estimate the state of a process with minimum error rate, Kalman filtering uses a set of mathematical equations. In 1970 Handschin, introduced Monte Carlo Localization (MCL), another new algorithm for mobile sensor localization. It is based on Markov localization; a family of probabilistic approaches. The purpose of this paper is to provide efficient and detailed information about the Kalman Filter, Extended Kalman Filter, Monte Carlo Localization, and Improved Monte Carlo Localization.*

Keywords*: Localization, Wireless Sensor Network, Kalman Filter, Extended Kalman Filter, Monte Carlo Localization (MCL), Improved Monte Carlo Localization (IMCL).*

1. INTRODUCTION

Wireless sensor network (WSN) is a collection of tiny sensing devices which are physically distributed over a wide area. These sensing devices are used to track the position of a moving target, to detect the presence of a contaminant in a water reservoir or to estimate the temperature in an orange grove etc…

2. KALMAN FILTER

Kalman filter is one of the most well-known and often-used mathematical tool that can be used for stochastic estimation from noisy sensor measurements. In 1960, Rudolph E. Kalman proposed a recursive solution to the discrete-data linear filtering problem and after that it was named as Kalman Filter. Kalman filtering is also known as linear quadratic estimation (LQE). It is a set of mathematical equations that uses a predictor-corrector type estimator to optimize the problem and minimizing estimated error covariance when some presumed conditions are met. In an environment which contains inaccuracies, noise (random variations), it uses a series of measurements observed over time and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone.

For any underlying system state, Kalman Filter uses recursive approach to produce optimized estimates. This algorithm is a two-step process-

- Prediction Step

In this step algorithm generates estimates of the current state variables along with their uncertainties.

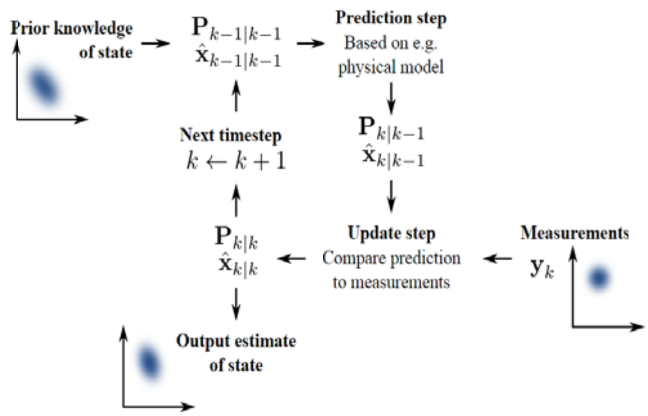
- Update Step

Once the result of the next measurement (necessarily corrupted with some amount of error, including random noise) is generated; these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty.

Kalman filter algorithm is recursive in nature; it uses the previously calculated state and its uncertainty matrix and present input to to produce optimized estimates. It does not require the additional past information. There are so many other extended and generalized methods (such as the extended Kalman filter and the unscented Kalman filter which work on nonlinear systems) of Kalman filter have been developed. Extended Kalman filter and unscented Kalman filter (the variants of Kalman filter) are the most celebrated and popular data fusion algorithms in the field of information processing. The most important use of Kalman filter was in the Apollo navigation computer that took Neil Armstrong to the moon, and (most importantly) brought him back. In recent years Kalman filters are at work in every smart phone, every satellite navigation device and many computer games

After 55 years, Kalman filter is one of the most important and common data fusion algorithms in use today, the purpose of using Kalman filter include -

- Smoothing noisy data and providing estimates of parameters of interest.
- Global positioning system receivers,
- Phase locked loops in radio equipment,
- Smoothing the output from laptop track pads, and many more.



The Kalman filter keeps track of the estimated state of the system and the variance or uncertainty of the estimate. The estimate is updated using a state transition model and measurements. $\hat{x}_{k|k-1}$ denotes the estimate of the system's state at time step k before the k-th measurement y_k has been taken into account; $P_{k|k-1}$ is the corresponding uncertainty.

A. Kalman Filter Algorithm

To estimate a process Kalman Filter uses a form of feedback control. At some time the filter estimates the process state and obtains feedback in the form of (noisy) measurements. There are two groups of equations for the Kalman filter.

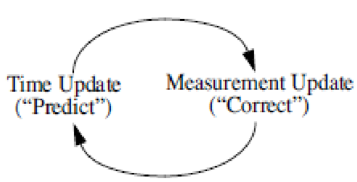
- Time Update Equations

To obtain the a priori estimates for the next time step, the current state and error covariance estimates are projecting forward (in time) by these equations. These time update equations is also known as predictor equations

- *Measurement Update Equations*

To obtain an improved a posteriori estimate, these equations incorporating a new measurement into the a priori estimate. These measurement update equations can also be known as corrector equations

In the below figure we resembles that of a predictor-corrector algorithm as the final estimation algorithm for solving numerical problems.



The figure shows the Kalman filter cycle. For projection of current state estimation we use the time update whereas these projected estimates are incorporated with actual measurement at that time by measurement update.

There are so many equations used by time update and measurement update which are given below-

For a discrete-time controlled process that is governed by the linear stochastic difference equation; The Kalman filter addresses the problem as below, to estimate the state x □□□^{n}-

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \quad (1)$$

Let Q be the process noise covariance and R be the measurement noise covariance matrices. The n □□n matrix A relates the state at the previous time step k – 1 to the state at the current step k. The n □□l matrix B relates the optional control input to the state x. The m □□n matrix H relates the state to the measurement z_k.

- Time update equations are

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (2)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (3)$$

- Measurement update equations are

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (4)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (5)$$

$$P_k = (I - K_k H)P_k^- \quad (6)$$

Here $\hat{x}_k^-$ □□□□^{n} is a priori state estimate at step k. P^-_k is the a priori estimate error covariance, P_k is the a posteriori estimate error covariance.

In measurement update initially we determine the Kalman gain, K_k. The next step is to actually measure the process to obtain z_k, and then by incorporating the measurement it generate an a posteriori state estimate (5). Posteriori error covariance estimate is the final step (6).

The process is repeated after each time and measurement update pair. The previous a posteriori estimates are used to project or predict the new a priori estimates with each repetition. This recursive feature of Kalman filter makes it very powerful algorithm. For example Wiener filter is designed to operate on all of the data directly for each estimate, whereas The Kalman filter recursively conditions the current estimate on all of the past measurements. This recursive-ness makes practical implementations of Kalman filter much more feasible.

3. EXTENDED KALMAN FILTER (EKF)

For a discrete-time controlled process, Kalman filter algorithm is basically used to estimate the state x □□□^{n}. This discrete-time controlled process should be governed by a linear stochastic difference equation. But the problem arises when the measurement relationship to the process is non-linear. The extended Kalman filter or EKF is the extension of the Kalman Filter that is used to linearize about the current mean and covariance.

Let us assume that our process is now governed by the non-linear stochastic difference equation and has a state vector x □□□^{n}. The non-linear stochastic difference equation is given by-

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{7}$$

The extended Kalman filter EKF is using so many time update and measurement update equations that are given below –

- Time update equations are-

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{8}$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{9}$$

The time update equations are used to project the state and covariance estimates from the previous time step k – 1 to the current time step k similar to basic discrete Kalman filter. Here Q_k is the process noise covariance at step k and A_k and W_k are the process Jacobians at step k.

- Measurement update equations are-

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{10}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)) \tag{11}$$

$$P_k = (I - K_k H_k) P_k^- \tag{12}$$

The measurement update equations are used to correct the state and covariance estimates with the measurement z_k similar to basic discrete Kalman filter. Here R is the measurement noise covariance at step k and H and V are the measurement Jacobians at step k.

4. MONTE CARLO LOCALIZATION

Monte Carlo localization (MCL) is a well-known algorithm which is used for position estimation of sensor nodes. In 1970 Handschin, introduced Monte Carlo methods and in 1993 Gordon, Salmond, & Smith reintroduced MCL for the target-tracking. This algorithm is also known as particle filter localization because it is using particle filters to estimate the position. Since a mobile sensor node moves and senses the environment, the Monte Carlo localization algorithm estimates the orientation and position of these moving nodes.

Particle filters are used by MCL algorithm to represent the distribution of likely states. These particles describe the hypothesis of where the sensor is that is these particles represent a possible state. Since at the beginning sensor have no information about its position, MCL uses the uniform random distribution of particles over the configuration space and assumes it is equally likely to be at any point in space. MCL predict new state of the sensor after its movement by shifting the particles. Recursive Bayesian estimation is used to resample the particles, whenever the sensor senses anything it means it correlate the actual sensed data with the predicted state. Finally with MCL algorithm the particles meet the actual position of the sensor. Monte Carlo Localization (MCL) uses fast sampling of the particle filters. To estimate the posterior distribution when the sensor moves or senses resampling is applied.

To determine the number of samples on-the-fly, Koller & Fratkina gave an adaptive sampling scheme in 1998 which is employed to trade-off computation and accuracy. Many samples used by the MCL during global localization. The size of the sample is small when the location of the sensor node is approximately known. There are many key advantages if sampling based representation is used; some of them are given below-

- It can globally determine the position of the sensor nodes and is able to multi-modal distributions in contrast to existing Kalman filtering based techniques.
- As compare to grid-based Markov localization, it reduces the amount of memory required and can integrate measurements at a higher frequency.
- Since state represented in the samples is not discretized, accuracy of Monte Carlo localization is more than the Markov localization with a fixed cell size.
- It implement is easier than other localization schemes.

A. Monte Carlo Localization Algorithm

The purpose of Monte Carlo Localization algorithm is to determine the position of the sensor nodes in an environment. As an input, this algorithm takes the data received from sensors z_t, an actuation command u_t, previous belief $X_{t-1} = \{x_{t-1}^{[1]}, x_{t-1}^{[2]}, \ldots, x_{t-1}^{[M]}\}$ at every time t and output of the algorithm is the new belief X_t.

Algorithm MCL (X_{t-1}, u_t, z_t)

$$\bar{X}_t = X_t = \emptyset$$

For m=1 to M

$x_t^{[m]}$ *=motion update(u_t, $x_{t-1}^{[m]}$)*

$w_t^{[m]}$ *=sensor update(z_t, $x_t^{[m]}$)*

$$\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$$

End for

For m=1 to M

Draw $x_t^{[m]}$ from $\bar{X}_t$ with probability proportional to $w_t^{[m]}$

$X_t = X_t + x_t^{[m]}$

End for

return X_t

5. IMPROVED MONTE CARLO LOCALIZATION

As discussed above Monte Carlo localization (MCL) is a technique for distance measurement of sensor nodes in wireless sensor network. It is based on particle filter combined with probabilistic models of sensor perception and motion. The main idea behind the MCL is to use a set of weighted samples and the posterior distribution of possible locations. There are two phase at each step-

Prediction phase: - In this phase when the sensor moves, the uncertainty of its position increases.

Update phase: - In this phase new observations are incorporated to filter and update data.

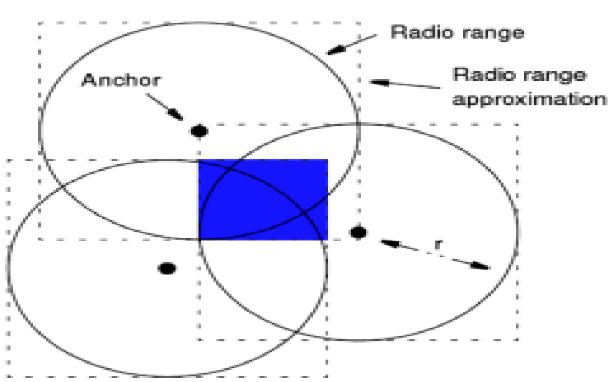
This is the recursive process where the sensor updates the predicted location continuously. There are two major problems with the previous existing schemes.

- First problem uses only a constant number of samples.
- Second is all nodes cannot ne localized in all time slots. Improved Monte Carlo Localization (IMCL) technique overcomes these problems. Network model is introduced

in IMCL scheme with five main parts that are discussed below as-

- Bounding-box construction
- Dynamic sampling
- Time series forecasting
- Samples weights computing
- Maximum possible localization error computing

- *Building the Bounding Box*



In Bounding box of Improved Monte Carlo Localization (IMCL) there are two areas involved. These areas are – The candidate samples area and the valid samples area.

Candidate Samples Area: - This area is used to draw new candidate samples.

Valid Samples Area: - This area is used to identify and filter out the invalid samples out of all the available samples.

There is a large probability that candidate samples drawn in the sampling step will be filtered out in the filtering step if the candidate samples area is large and the valid samples area is small. The construction of bounding box in IMCL schemes is shown above-

- Dynamic sampling method

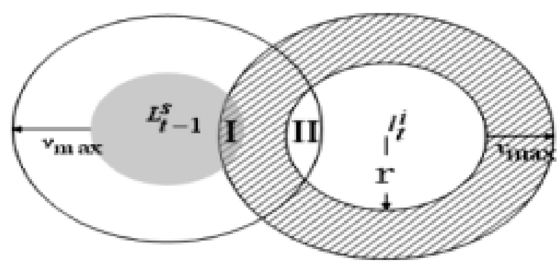
For localization, the number of samples in the previous schemes is fixed like 30 or 50. But in Improved Monte Carlo scheme the number of samples is dynamic that is based on the size of the sample area. To estimate the nodes location accurately for a large anchor box, a large number of samples are needed. Whereas we focuses on a small area if the anchor box is small and for that small area to determine the position accurately; a small number of samples is needed.

- Linear prediction using time series

In a time- varying environment to predict time series, linear prediction method is a very helpful and powerful technique.

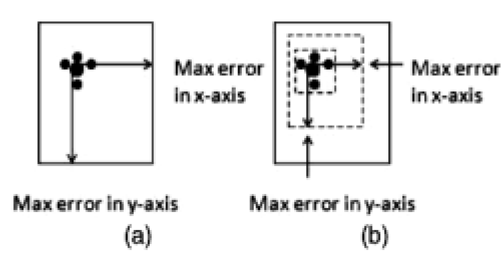
- Weighting the Samples

When a candidate sample is chosen, its weight is being calculated by using 1-hop (anchor) and 2-hop (common) neighbor nodes. The technique of computing weight is given below-



- Error Computation

The sensor node determines its position by computing the weighted average of these samples after obtaining N valid samples. Also with the help of the bounding box and position estimations, the sensor node can also determine the ERx and ERy as shown below.



6. CONCLUSION

In this paper we discussed the problem of localization in wireless sensor network and studied so many approaches like Kalman Filtering, Extended Kalman Filtering, Monte Carlo Localization (MCL), and Improved Monte Carlo Localization (IMCL) to overcome this issue of localization in WSN. This paper will help to wireless sensor network designer to choose the best method (among the discussed above), to implement in their applications.

REFERENCES

[1] R. Manzoor, Energy efficient localization in wireless sensor networks using noisy measurements [M.S. thesis], 2010.

[2] Hu, L and Evans, D "Localization for Mobile Sensor Networks,"Proc. ACM MobiCom, pp. 45-57.2004.

[3] G. Sarigiannidis, "LOCALIZATION FOR AD HOC WIRELESS SENSOR NETWORKS",M.S. thesis TECHNICAL UNIVERSITY DELFT, THE NETHERLANDS, AUGUST 2006

[4] Kalman, R. 1960. A new approach to linear filtering and prediction problems. Tansaction of the ASME–Journal of basic engineering 35–45.

[5] Baggio and Langendoen, K "Monte-Carlo Localization for Mobile Wireless Sensor Networks", Proc. Conf. Mobile Ad-Hoc and Sensor Networks (MSN '06), pp. 317-328.2006.

[6] B. Soltaninasab and J.Amiri, "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, 2010.

[7] Kanazawa, K.; Koller, D.; and Russell, S. 1995. Stochastic simulation algorithms for dynamic probabilistic networks. Proc. Of UAI-95.

[8] Kitagawa, G. 1996. Monte carlo filter and smoother for nongaussian nonlinear state spacemodels. Journal of Computational and Graphical Statistics 5(1).

[9] Koller, D., and Fratkina, R. 1998. Using learning for approximation in stochastic processes. Proc. of ICML-98.

[10] Kortenkamp, D.; Bonasso, R.; and Murphy, R., eds. 1997. AIbased Mobile Robots: Case studies of successful robot systems. MIT Press.

[11] Fox, D. 1998. Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Naviagation. Ph.D. Diss, University of Bonn, Germany.

[12] Gelb, A. 1974. Applied Optimal Estimation. MIT Press.

[13] Gutmann, J.-S., and Schlegel, C. 1996. AMOS: Comparison of scanmatching approaches for self-localizati on in indoor environments. Proc. of Euromicro. IEEE Computer Society Press.

[14] Gutmann, J.-S.; Burgard, W.; Fox, D.; and Konolige, K. 1998. An experimental comparison of localization methods. Proc. Of IROS-98.

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

[16] Isard, M., and Blake, A. 1998. Condensation–conditional density propagation for visual tracking. International Journal of Computer Vision 29(1).

[17] Kalman, R. E. 1960. "A New Approach to Linear Filtering and Prediction Problems," Transaction of the ASME—Journal of Basic Engineering, pp. 35-45 (March 1960).

[18] Handschin, J. 1970. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. Automatica 6.

[19] Lewis, Richard. 1986. Optimal Estimation with an Introduction to Stochas-tic Control Theory, John Wiley & Sons, Inc.

[20] Gordon, N.; Salmond,D.; and Smith, A. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Procedings F 140(2).

[21] Maybeck, Peter S. 1979. Stochastic Models, Estimation, and Control, Vol-ume 1, Academic Press, Inc.

[22] Boukerche .A, H. Oliveira, E. Nakamura , and A. Loureiro, "Secure localization algorithms for wireless sensor networks," IEEE Commun. Mag., vol. 46, no. 4, pp. 96-101.2008.

[23] H. Shi, X. Li, Y. Shang, D. Ma "Cramer-Rao Bound Analysis of Quantized RSSI Based Localization in Wireless Sensor Networks", In Proc. Of International Conference on parallel and distributed systems (ICPAD'05), July 2005.

[24] Sorenson, H. W. 1970. "Least-Squares estimation: from Gauss to Kalman," IEEE Spectrum, vol. 7, pp. 63-68, July 1970.

[25] Brown, R. G. and P. Y. C. Hwang. 1992. Introduction to Random Signals and Applied Kalman Filtering, Second Edition, John Wiley & Sons, Inc.

[26] Datta, S. Klinowski, C. Rudafshani, M. and Khaleque, M "Distributed Localization in Static and Mobile Sensor Networks,"Proc. IEEE Int'l Conf. Wireless and Mobile Computing, pp. 69-76.2006.