The 5th International Conference on
Engineering Mechanics and Automation (ICEMA 5)
Hanoi, October 11÷12, 2019

Employing Extended Kalman Filter with Indoor Positioning System for Robot Localization Application

Dong Tran Huu Quoc[a], Anh Phan Hoang[a], Tuan Nguyen Dinh[a], Nam Bui Duy[a], Van Nguyen Thi Thanh[a], Tung Bui Thanh[a], Trinh Chu Duc[a]

[a] VNU - University of Engineering and Technology, Hanoi, 123106, Vietnam
trinhcd@vnu.edu.vn

Abstract

IPS (Indoor Positioning System) is used to localize the robot position in a narrow environment. However, IPS signal is not accurate in some cases, the IPS signal transmission will be obstructed by people the public museums, for example. The combination with other sensors will reduce the poor signal from IPS and noise in the environment with many obstacles. The Extended Kalman Filter (EKF) have been widely used for mobile robot localization system and gained certain results. In this paper, EKF is embedded to Central Processing Unit with Robot Operating System on "UET-FuSo" robot to fuse IPS signal with encoder and IMU sensor for the determination of position and orientation robot. The obtained results show the effect of the proposed method when the robot localization is more stable and accurate than using only IPS signal. This localization will be applied for mapping and navigation of mobile robot in exhibition guidance robot.

Key Words: Extended Kalman Filters, IPS, Localization, ROS

1. Introduction

In progress working with Robot on real environment and real situation, it's positions are required conditions, without them Robot can't locate and move. IPS is used to get Robot coordinates for localization on small or medium places that GPS and other satellite technologies lack of precision or fail entirely, such as inside multistory buildings, airports, alleys, parking garages, and underground locations. However, IPS doesn't always give accurate signal, especially if so many people obstructing the transmission. Moreover, not only IPS but also sometimes sensor data isn't accurate and have so much noise. To make Robot more smart and doesn't do anything jerkily, stable messages and trustful data is needed. Therefore, the most reasonable way is combine all sensors and make each sensor support another. To solve these problems, Kalman Filters is considered. On the other hand, if the robot has enough sensors fuse with IPS for Kalman Filters, it feel everything around very quickly and accurately. Filtering is a very common method in engineering and embedded system, especially in Robot.[1] A good filtering algorithm can reduce the noise from signals while retaining the useful data.[3] Kalman Filters is one of the common filters use for Signal Processing.[6][10]

2. Indoor Positioning System

Indoor Positioning System is an off-the-shelf indoor navigation system, designed to provide precise location data to autonomous robots, vehicles (AGV), and copters. It can also be used to track moving objects via mobile beacons attached to them. The navigation system consists of a network of stationary ultrasonic beacons interconnected via radio interface in a license-free band, one or more mobile beacons installed on objects to be tracked and modem providing gateway to the system from a PC or other computers. Mobile beacon's location is calculated based on a propagation delay of an ultrasonic pulses (Time-Of-Flight or TOF) between stationary and mobile beacons using trilateration algorithm. The system can build the map of stationary beacons automatically (For Non-Inverse Architecture).

Minimum configuration requirements (Non-Inverse Architecture) to ensure optimal performance of the Indoor Positioning System:

- For 3D (X, Y, Z) tracking: an unobstructed line of sight (hearing) between a mobile beacon and 3 or more stationary beacons within 30 meters

- For 2D (X, Y) tracking: an unobstructed line of sight (hearing) between a mobile beacon and 2 or more stationary beacons within 30 meters

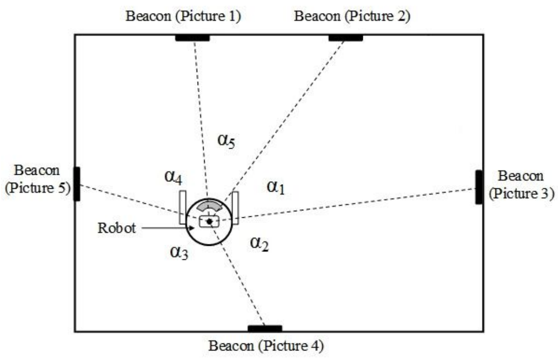

Figure 1. Robot Get Positions with IPS Beacons

We built a program to get 2D coordinates (X, Y) from IPS. After that, 2D coordinates is used for localization, used for Robot navigation. If the goals to Robot are sent, system calculated a path to go to goals for a Robot and send velocities to motors. Robot stopped when it reached the goals, coincides with robot coordinates get from IPS. But these are requirements for optimal performance. In real situation, data isn't accurate if a line of sight from beacon to beacon and from beacon to robot is obstructed. People walking through the lines make coordinates got from IPS is jerky.

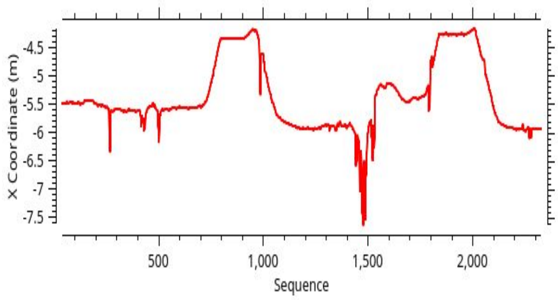

Figure 2. Raw X Coordinate/Sampling Times from IPS

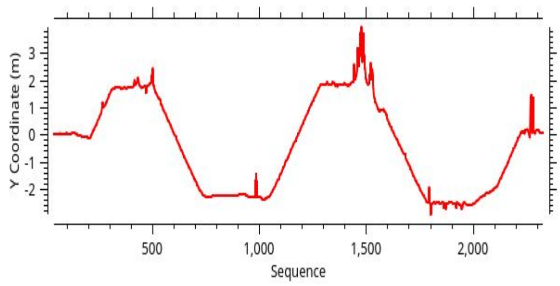

Figure 3. Raw Y Coordinate/Sampling Times from IPS

Figure 2 and 3 above represent X and Y coordinates on the domain of the sampling times.

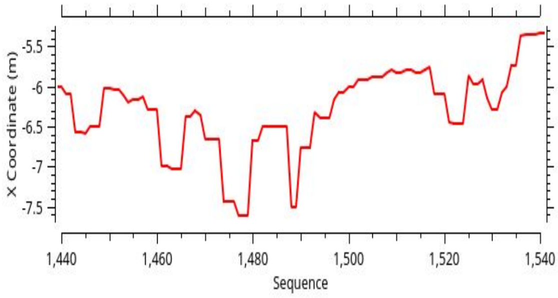

Figure 4. Raw X Coordinate with low accuracy

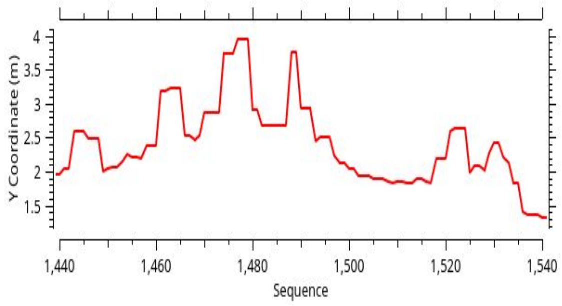

Figure 5. Raw Y Coordinate with low accuracy

It can be seen in figures 4 and 5 that the coordinates fluctuate sharply, from about 1440th to 1540th sampling. So robot wasn't stable and moved jerkily. Signals are raw and didn't go through any signal processing system. These noises appeared because so many people walking around our robot and they obstructed the radio line.

Extended Kalman filters (EKF) can decrease them. So we embed EKF to system, combined with Encoder and Imu, to make signals great again.

3. Extended Kalman Filter Implementation

The Extended Kalman filter (EKF) is a mathematical tool that can estimate the variables of a wide range of process to nonlinear systems. It works by linearizing the nonlinear state dynamics and measurement models. It is widely used in robot engineering, popular in navigation, positioning and control applications.[12][13]

This type of filter works very well in practice and that is why it is often implemented in embedded control system and because robot needs an accurate estimate of the process variables. The Extended Kalman filter is a smarter way to integrate measurement data into an estimate by recognising that measurements are noisy and that sometimes they should be ignored or have only a small effect on the state estimate. It smooths out the effects of noise in the state variable being estimated by incorporating more information from reliable data than from unreliable data. The user can tell the Extended Kalman filter how much noise there is in the system and it calculates an estimate of the position taking the noise into account[5][14] Extended Kalman Filter algorithm is still the most basic and common solution for discrete and low accurate signals such as GPS, IPS,...[7]

In our robot - "UET-FuSo", IPS, Wheel Encoders and IMU are fused into EKF and embed to Central Processing Unit for localization part. The input is 2D data format includes: Coordinates from IPS (MarvelMind Beacon Indoor Position System), Velocities from Wheel Encoders, Rotations and Angular Velocities measured by Gyroscope sensor, Accelerations measured by Accelerometer sensor, 2 sensors are integrated into a 6DOF - Sensor called IMU-MPU6050.

The EKF algorithm is divided into three parts: Initialization and Linearization, Prediction and Update. Assume that robot had x y coordinate from IPS, v_x v_y linear velocity from Wheel Encoders, a_x a_y yaw v_{yaw} linear acceleration, orientation and angular velocity from IMU. Our goal is to predict, update and process 2D coordinates for robots. so the kinematic and kinetic equations for the mobile robot:

$$\widehat{x_k} = \begin{bmatrix} x_k \\ v_{x_k} \\ a_{x_k} \end{bmatrix} = \begin{bmatrix} x_{k-1} + \Delta t \cdot v_{x_{k-1}} + \frac{1}{2} \cdot \Delta t^2 \cdot a_{x_{k-1}} + w_{1_{k-1}} \\ v_{x_{k-1}} + \Delta t . a_{x_{k-1}} + w_{2_{k-1}} \\ a_{x_{k-1}} + w_{3_{k-1}} \end{bmatrix} \quad (1)$$

$$\widehat{x'_k} = \begin{bmatrix} x'_k \\ v'_{x_k} \\ a'_{x_k} \end{bmatrix} = \begin{bmatrix} x_k + u_{1_k} \\ v_{x_k} + u_{2_k} \\ a_{x_k} + u_{3_k} \end{bmatrix} \quad (2)$$

All equations are implemented to get updated x_k, calculate y_k is similar.

The relationship of each state to the previous state is shown as follows:

$$x_k = f(x_{k-1}) + w_{k-1} \quad (3)$$

Where x_k is the state parameter of signal at time k, f is a nonlinear function representing x_k from x_{k-1}.

In addition, eight parameters including coordinates, linear velocities, linear accelerations, rotation angles and angular velocities are represented by 3 vectors. Because the units and scale of sensor data might not be the same as the units and scale of the measurement. So the state to measurement systems transform function is represented:

$$x'_k = h(x_k) + u_k \tag{4}$$

Where x_k is the status parameter of signal at time k, h is a function mapping signal into the measurement space containing x_k signal.

- w_{k-1} and u_k respectively are process and observation noises. These two types of noise have multivariate Gaussian form with covariances, Q and R respectively.

- f and h can be used to calculate and predict the state of a signal. However, if using these two functions, the robot will not be able to use the covariances to estimate. Jacobi matrices is used instead with partial derivative to calculate. Let F and H respectively be the Jacobi matrices of partial derivatives of f and h, with respect to x_k, F and H have the form:

$$F = \left.\frac{\partial f}{\partial x}\right|_{\widehat{x_{k-1}},\, w_{k-1}} \tag{5}$$

$$H = \left.\frac{\partial h}{\partial x}\right|_{\widehat{x_k}} \tag{6}$$

$$\widehat{x_k} = F.\widehat{x_{k-1}} + w_{k-1} \tag{7}$$

$$\widehat{x'_k} = H.\widehat{x_k} + u_k \tag{8}$$

With $w_{k-1} = 0$, $u_k = 0$, the Jacobian Matrices F, H are:

$$F = \begin{bmatrix} 1 & \Delta t & \Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

This is an estimation of the current covariance over time, represented by equation, Predicted Covariance Estimate:

$$P'_k = F.P_{k-1}.F^T + Q \tag{11}$$

P_{k-1} is the covariance matrix of signals at time k - 1, F^T is the transposed matrix of F, Q is the matrix representing the noise during the estimation process, set by hand.

Particularly for the first signal received, since there is no previous signals to calculate the state, the matrix P_0 will be set manually. This is also the first initialization step.

$$P'_1 = F.P_0.F^T + Q \tag{12}$$

Matrix P'_k will be included to update the new status for the signal, we employ the Joseph form covariance update equation to promote filter stability by ensuring that P_k remains positive semi-definite[2]. Specific steps are as follows:

First, compute the Kalman Gain:

$$K = P'_k.H^T\left(H.P'_k.H^T + R\right)^{-1} \tag{13}$$

Second, update new state estimate:

$$x_k = \widehat{x_k} + K\left(\widehat{x'_k} - H.\widehat{x_k}\right) \tag{14}$$

Last, update new covariance estimate:

$$P_k = (I - K.H)P'_k(I - K.H)^T + K.R.K^T \tag{15}$$

In particular, H^T is the transposed matrix of H, R is covariance matrix representing signal noises of data from the sensors. We also write a program to calculate R with equation (16):

$$R = \begin{bmatrix} \Sigma_x & 0 & 0 \\ 0 & \Sigma_{v_x} & 0 \\ 0 & 0 & \Sigma_{a_x} \end{bmatrix} \tag{16}$$

$$\sum_x = \frac{1}{k-1}.\sum_{i=1}^{k}(x_i - \bar{x})^2 \tag{17}$$

Set the process noise covariance Q by hand is difficult. A problem is that u_k and w_{k-1} are uncorrelated zero-mean Gaussian noise sequences with covariance matrix Q and R[8]. So that EKF doesn't do the best work in theory.

Final, get and use x, y coordinates after update step instead of raw data from IPS.

4. Program Algorithm Implementation

- First sampling:

- Input:

1.1. The first signal received gives coordinate x_0 y_0, velocity v_{x0} v_{y0}, acceleration a_{x0} a_{y0}, Q and P_0 set by hand

1.2. Coordinate x_1 y_1 is the x y coordinate at time of first sampling (k = 1)

1.3. Velocity v_{x1} v_{y1} is the x y velocity at time of first sampling (k = 1)

1.4. Acceleration a_{x1} a_{y1} is the x y acceleration at time of first sampling (k = 1)

- Calculate:

1.5. Calculate F and H using the formula (7) (8)

1.6. The filter will calculate covariance P'_1 from P_0 and Q using the formula (12)

1.7. Calculate Kalman Gain (K) using the formula (13)

- Output:

1.8. Update x_1 y_1, using the formula (14)

1.9. Update Covariance P_1, using the formula (15)

- Second sampling:

- Input:

2.1. The EKF gives coordinate x_1 y_1, velocity v_{x1} v_{y1}, acceleration a_{x1} a_{y1}, Q and P_1

2.2. Coordinate x_2 y_2 is the x y coordinate at time of second sampling (k = 2)

2.3. Velocity v_{x2} v_{y2} is the x y velocity at time of first sampling (k = 2)

2.4. Acceleration a_{x2} a_{y2} is the x y acceleration at time of first sampling (k = 2)

- Calculate:

2.5. Calculate F and H using the formula (7) (8)

2.6. The filter will calculate covariance P'_2 from P_1 and Q using the formula (11)

2.7. Calculate Kalman Gain (K) using the formula (13)

- Output:

2.8. Update x_2 y_2, using the formula (14)

2.9. Update Covariance P_2, using the formula (15)

- Third sampling and more: Same as Second sampling

5. Experimentations and Results

In this section, x, y coordinates provided by raw data obtained from IPS will be compared with x, y coordinates after using EKF. For this purpose both signals were implemented in rqt_multiplot setup on ROS (Robot Operating System)[9], a tool help us to plot all messages got in embedded system into coordinate systems. We control a Robot to goal points with a very simple case: the robot that follows a path obtained from the system model. In the figures it is presented the estimated path of the robot compared to the real path.

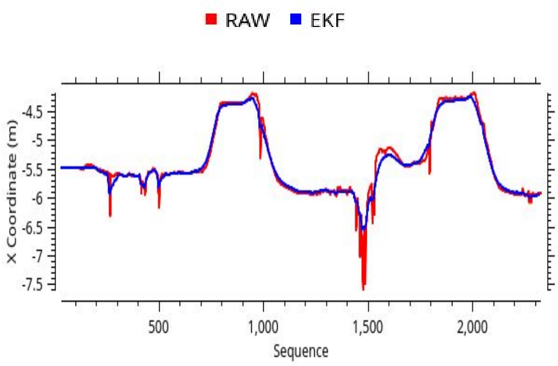

Figure 6. Illustration of using the EKF to estimate the position on the X axis

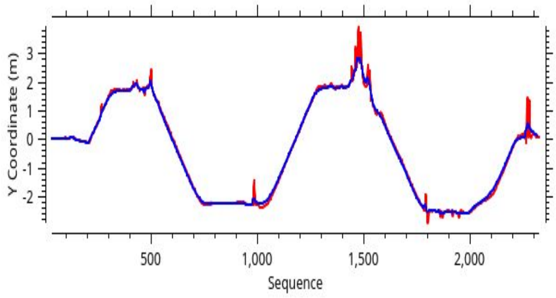

Figure 7. Illustration of using the EKF to estimate the position on the Y axis

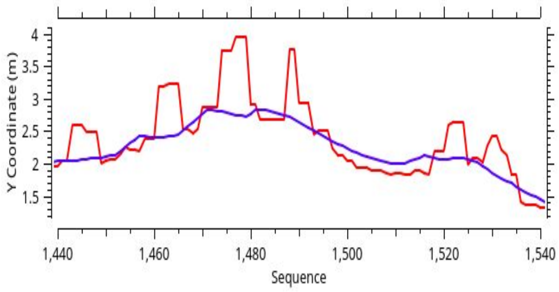

Figure 8. Using the EKF to estimate position on the X axis with low accuracy

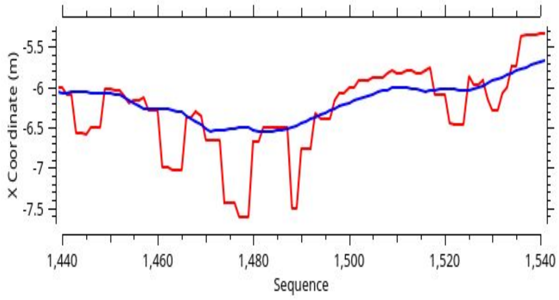

Figure 9. Using the EKF to estimate position on the Y axis with low accuracy

In figure 6, it is shown the path estimated with the help of the EKF and the raw path get from IPS on X axis. The same plot shows estimated data for Y axis shown in figure 7. In both plots it can be seen clearly that the EKF (blue line) predicted path is closer to the real path (red line). But at about 1440^{th} to 1540^{th} sampling or other sampling, IPS send us very poor data but EKF filter data is better (Figure 8 and 9). The difference between the performance of the filter and raw can be seen above.

6. Conclusions And Future Works

In this paper, the Indoor Positioning System with Extended Kalman filters is embedded into a real robot system to localize robot in special environment. The results show that the fusion between IPS and other sensors in EKF makes the robot localization more stable and accurate than the only IPS usage.

In order to implement the simultaneous localization and mapping (SLAM)[11] next time, the camera data such as visual coordinates, velocities, and orientations will be fused into the EKF. This will make the estimation of robot state more accurate.

7. Acknowledgement

This work has been supported by Vietnam National University, Hanoi (VNU), under Project No. QG.17.69.

8. References

[1] Aguirre, E., González, A., Muñoz, R. (2004), *Mobile Robot Map-Based Localization using approximate locations and the Extended Kalman Filter*, Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Perugia, Italy, pp. 191-198.

[2] Bierman, G. J. and Thornton, C. L. (1977), *Numerical comparison of Kalman filter algorithms: Orbit determination case study*, Automatica vol. 13, no. 1, pp. 23-35.

[3] Casanova, O. L., Alfissima, F., Machaca, F. Y. (2008), *Robot Position Tracking Using Kalman Filter*, Proceedings of the World Congress on Engineering, London, UK, pp. 1604-1608.

[4] Choomuang, R., Afzulpurkar, N. (2005), *Hybrid Kalman Filter/Fuzzy Logic based Position Control of Autonomous Mobile Robot*, International Journal of Advanced Robotic Systems, IN-TECH, Vol. 2, No. 3, pp. 197-208.

[5] Kalman, R. E. (1960), *A New Approach to Linear Filtering and Prediction Problems*. Transactions of the ASME pp. 35-45, 1960.

[6] Kim, S. G., Crassidis, J. L., Cheng, Y., Fosbury, A. M., Junkins, J. L. (2007), *Kalman filtering for relative spacecraft attitude and position estimation*, Journal of Guidance Control and Dynamics, American Institute Of Aeronautics and Astronautics, Vol. 30, No. 1 pp. 133-143.

[7] Leonard, J. J. and Durrant-Whyte, H. F. (1991), *Mobile robot localization by tracking geometric beacons*, Robotics and Automation, IEEE Transactions on vol. 7, no. 3, pp. 376-382.

[8] Odelson, B. J., Rajamani, M. R., Rawlings J. B. (1997), *A new autocovariance least-squares method for estimating noise covariances*, Automatica vol. 13, no. 1, pp. 23-35.

[9] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A.Y. (2009), *ROS: an open-source robot operating system*, ICRA workshop on open source software vol. 3, no. 3.2.

[10] Santana, A. M., Sousa, A. S. S., Britto, R. S., Alsina, P. J., Medeiros, A. A. D. (2008), *Localization Of A Mobile Robot Based In Odometry And Natural Landmarks Using Extended Kalman Filter*, Proceedings of the 5th International Conference on Informatics in Control, Automation & Robotics, Funchal-Madeira, Portugal.

[11] Sasiadek, J.Z. Monjazeb, A. Necsulescu, D. (2008), *Navigation of an autonomous mobile robot using EKF-SLAM and FastSLAM*, Proceedings of the 16th Mediterranean Conference on Control and Automation, Ajjacio, France, pp. 517-522.

[12] Thomas, M., Daniel, S. (2015), *A Generalized Extended Kalman Filter Implementation for the Robot Operating System*, Intelligent Autonomous Systems 13 Proceedings of the 13th International Conference IAS-13.

[13] Wang, L. C., Yong, L. S., Ang, M. H. (2002), *Mobile Robot Localisation for Indoor Environment*, Technical Report, Singapore Institute of Manufacturing Technology, Singapore.

[14] Welch, G., Bishop, G. (2006), *An Introduction to the Kalman Filter*, Technical Report, University of North Carolina, Chapel Hill, USA.