Accurate Navigation for Autonomous Mobile Robot by Extended Kalman Filter and Indoor Global Positioning System

Jaeyong Park, Jungwon Choi, Kwangwon Park, Sukgyu Lee, Juhyun Park
Yeungnam University, Korea
Daedong 214-4, Gyoungsan, Gyoungbuk, Korea
announcer4@hotmail.com, consys@korea.com, zeromos@hotmail.com, sglee@yu.ac.kr, rcl@yu.ac.kr

Abstract - The purpose of this paper is to improve accuracy of robot’s navigation. It is difficult to estimate robot’s position by only odometry. To correct robot’s position uncertainty, we adapt the extended kalman filter to the robot. And to enhance accuracy of robot navigation, we build an indoor global positioning system.

1. Introduction

The research through the robot is extended up to the industrialization and the home service robot with regard to the localization and the map building, a variable investigation is followed. Navigation ability is basic and essential to recognize exactly unknown environment for autonomous mobile robot system.

However, this problem has not been solved yet even though there were many researches. Because robot must not only estimate own position exactly to get accurate map, but also construct accurate map to estimate exactly own position. In other words, that is the chicken-or-egg relationship.

The kalman filter is useful for robot navigation. But the basic kalman filter is limited to a linear assumption. However, most non-trivial systems are non-linear. The non-linearity can be associated either with the process model or with the observation model or with both. So they need linearization by partial derivatives.

In this paper, to develop an autonomous mobile robot which is able to apply to the general environment and make sure the feasibility, the position information about the indoor structure environment by using the global ultrasonic sensor is transmitted to the robot. And to get the information about the obstacles, the robot has five ultrasonic sensors and infrared sensors with 45 degrees. The infrared sensors are used to correct the error about the feasibility of reflection when the robot recognizes a obstacle.

The paper is organized as follows. In the next section, the navigation problem is briefly reviewed, and then in section 3 the basic kalman filter method is introduced. Section 4 describes a extended kalman filter and section 5 provides a simulation results. A conclusion and future work are presented in section 6 and 7 respectively.

2. The navigation Problem

In most cases, when robot estimates own position, it is calculated by the odometry. However the odometry information is corrupted by the systematic or non-systematic error. Therefore robot can’t estimate position exactly. As the odometry error is accumulated, robot can not construct exact map.

So robot needs some algorithm that correct accumulated odometry error. The kalman filter can correct it through observation.

In robot navigation, a state vector contains the pose of the robot (its location and orientation, \( x, y, \theta \)) relative to its environment, along with the location of landmarks in the robot’s proximity. In what follows, we will represent the vector of state variables at time \( t \) by \( x_t \). Navigation addresses situations in which state variables are not observable directly. In such situations, the robot has to rely on information obtained from sensor and robot motion.

\[
\begin{align*}
x_{t+1} &= Fx_t + Gu_t + v_t \\
z_{t+1} &= Hx_{t+1} + m_t
\end{align*}
\]

In the above equations \( F, G \) and \( H \) are matrices; \( t \) is the time index; \( x \) is called the state of the system; \( u \) is a known input to the system; \( z \) is the measured output; and \( v \) and \( m \) are the noise. The variable \( v \) is called the process noise, and \( m \) is called the measurement noise. Each of these quantities is (in general) vector and therefore contains more than one element. The vector \( x \) contains all of the information about the present state of the system, but we cannot measure \( x \) directly. Instead we measure \( z \), which is a function of \( x \) that is corrupted by the noise \( v \). We can use \( z \) to help us obtain an estimate of \( x \), but we cannot necessarily take the information from \( z \) at face value because it is corrupted by noise. The measurement is like a politician. We can use the information that it presents to a certain extent, but we cannot afford to grant it our total trust.

3. Basic Kalman Filter

The Kalman filter is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements.

The Kalman filter has three distinct phases: predict measurement and update. The predict phase uses the state estimate from the previous time step to produce an estimate of
the state at the current time step. The measurement phase is that robot observes fixed object or feature thing to correct own position. In the update phase, measurement information at the current time step is used to refine this prediction to arrive at a new, (hopefully) more accurate state estimate, again for the current time step.

![Figure 1: Localization algorithm using kalman filter.](image)

Many real dynamical systems do not exactly fit this model. However, because the kalman filter is designed to operate in the presence of noise, an approximate fit is often good enough for the filter to be very useful. Variations on the kalman filter described below allow richer and more sophisticated models.

### 4. Extended Kalman Filter

The basic kalman filter is limited to a linear assumption. However, most non-trivial systems are non-linear. The non-linearity can be associated either with the process model or with the observation model or with both. In the extended kalman filter (EKF) the state transition and observation models need not be linear functions of the state but may instead be (differentiable) functions.

In two dimensional environment, the robot’s state vector \( \mathbf{x} \) has \( x \) position, \( y \) position and \( \theta \) heading. The robot is driven by two wheels and steer wheel, the state variation can describe as followed equation, through present state and control input.

\[
x_{k+1} = f(x_k, u_k, v_k)
\]

\[
x_{(k+1)} = x_k + T \frac{r_y W_{R(k)}}{2} \cos\theta_{(k)} + \frac{r_x W_{L(k)}}{2} \sin\theta_{(k)}
\]

\[
y_{(k+1)} = y_k + T \frac{r_y W_{R(k)}}{2} \cos\theta_{(k)} + \frac{r_x W_{L(k)}}{2} \sin\theta_{(k)}
\]

\[
\theta_{(k+1)} = \theta_k + T \frac{r_y W_{R(k)}}{2} - \frac{r_x W_{L(k)}}{2}
\]

To linearize, a matrix of partial derivatives (the Jacobian) is computed.

\[
F_{(k)} = \nabla_x f = \begin{bmatrix}
1 & 0 & -\frac{T r_y W_{R(k)}}{2} + \frac{T r_x W_{L(k)}}{2} \sin\theta_{(k)} \\
0 & 1 & \frac{T r_y W_{R(k)}}{2} + \frac{T r_x W_{L(k)}}{2} \cos\theta_{(k)} \\
0 & 0 & 1
\end{bmatrix}
\]

![Figure 2: Configuration of sensor observation.](image)

We can make observation model through the figure 2.

\[
z_t = h(x_t, m_t)
\]

\[
P_r = \begin{bmatrix}
x + l \cos\theta \\
y + l \sin\theta \\
z_c
\end{bmatrix},
P_f = \begin{bmatrix}
x - l \cos\theta \\
y - l \sin\theta \\
z_c
\end{bmatrix}
\]

The function \( f \) can be used to compute the predicted state from the previous estimate and similarly the function \( h \) can be used to compute the predicted measurement from the predicted state. However, \( f \) and \( h \) cannot be applied to the covariance directly.

To apply to the EKF using already explained system model, first, the predictions of the state and the covariance are computed by:

\[
\hat{x}_{t+1|T} = f(\hat{x}_t, u_t)
\]

\[
P_{t+1|T} = FP_{t+1|T} + QRT
\]

Once we have the measures of the distance to the beacon, the innovation can be computed by:

\[
\hat{y}_{t+1} = z_{t+1} - h(\hat{x}_{t+1|T})
\]

and the covariance of the innovation by:

\[
S_{t+1} = HP_{t+1|T} H^T + R_{t+1}
\]

a matrix \( H \) is the jacobian matrix of observation model. After that, the gain of the filter can be computed by:

\[
K_{t+1} = P_{t+1|T} H^T S_{t+1}^{-1}
\]

and the estimation of the state of the system and the covariance of the finally obtained by:

\[
\hat{x}_{t+1} = \hat{x}_{t+1|T} + K_{t+1}(z_{t+1} - h(\hat{x}_{t+1|T}))
\]

\[
P_{t+1} = (I - K_{t+1} H) P_{t+1|T}
\]
5. Simulation Results

The performance of the two types, using an indoor GPS and non-using indoor GPS, were compared by simulations. Figure 3 shows the effect of the odometry error. Without re-localization, the robot localization error increases in an unbounded fashion. But the robot estimates localization well in figure 4 compare with figure 3.

\[
\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + K_{t+1}y_{t+1}, \\
\hat{p}_{t+1|t+1} = \hat{p}_{t+1|t} - K_{t+1}S_{t+1}K_{t+1}^T.
\]

The real robot trajectory by odometry error corrupts obstacle position in figure 3. In other words, that is a correlation between robot path error and map error. But in figure 4, as the robot corrects pose error by an indoor GPS, the robot can estimate own position and build obstacles exactly.

6. Conclusion

The robot navigation algorithm has been outlined in this paper. Preliminary simulation results in a small area for this method have been obtained in real time, and compare with results obtained using an indoor GPS. In this paper we could get the result of a more accuracy robot’s navigation using by the extended kalman filter and an indoor global positioning system. As a robot moves, uncertainty of robot’s pose becomes high by systematic error and non-systematic error., but it can be corrected through the extended kalman filter and an indoor global positioning system.

7. Future Work

Future work on this project will test performance by a real robot, and will test an experiment over a large area. Also when the initial robot’s location is unknown or the robot becomes completely lost, the study of other approaches to carry out the probabilistic transformation. To apply to an outdoor environment, the system could also be coupled with the natural landmark system. These researches can be associated with SLAM problems.

References


