Chapter from the book
Downloaded from: http://www.intechopen.com/books/

Interested in publishing with InTechOpen?
Contact us at book.department@intechopen.com
Correction Robot pose for SLAM based on Extended Kalman Filter in a Rough Surface Environment

Jaeyong Park, Sukgyu Lee, Joohyun Park
Electrical Engineering Department, Yeungnam University, Korea
Electrical Engineering Department, Yeungnam University, Korea
Electrical Engineering Department, Yeungnam University, Korea
announcer4@hotmail.com

Abstract: This research deals with mobile robot SLAM algorithm based on extended kalman filter. To enhance a accuracy of robot pose, one more extended kalman filter is used in a rough surface environment. The robot has uncertain kinematic model due to a caterpillar. When the robot drives on irregular surface, it’s heading can be corrupted. We propose a method to correct uncertain robot pose using one more extended kalman filter through simulation results.

Keywords: SLAM(Simultaneous lacaization and mapping), EKF, Uncertain robot pose, Dynamics of caterpillar.

1. Introduction

It is a basic and essential ability to recognize exactly unknown environment for intelligent 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 relation between localization and mapping is a consequence of how errors in the robot’s sensor readings are corrupted by error in robot’s pose. As the robot moves, its pose estimate is corrupted by motion noise. The perceived locations of obstacles in the environment are corrupted by both measurement noise and the error in the estimated pose of the robot. However unlike measurement noise, error in the robot’s pose will have a systematic effect on the error in the map. In general, this effect can be stated more plainly. As a result, the true map cannot be estimated without also estimating the true pose of robot.

A robot can be classified by drive method. The most general method that is used to a robot is a differential type using two wheels. For the residue, there are omnidirectional wheel, caterpillar, four drive wheel. The differential type using two wheel is one the easiest type to make a motion model and consider motion error compare with others type of robot, but it is constrained an environment to drive. We assume that the drive type of robot is a caterpillar type in this simulations. The caterpillar type robot can implement SLAM in a rough surface environment compare with a differential wheel type robot. However because of the nature of a caterpillar, the friction surface is wide, slip, ambiguity of kinematic analysis, we can not make its motion model easily. Also when the robot drives on a rough surface road, as its heading is corrupted, estimation of robot’s pose has difficulty.

This paper introduces a useful method for correcting robot pose by using one more extended kalman filter. The robot implements SLAM based on the extended kalman filter, one more extended kalman filter is used to supplement uncertain robot pose due to unreliable robot’s kinematic property and to correct rocking robot’s heading because of irregular driving surface.

The paper is organized as follow. In the next section, the SLAM based on EKF is briefly reviewed, and then in section 3 two kinds of robot’s motion models are compared. Section 4 proposes the method for correcting corrupted robot’s heading using another EKF and section 5 provides the simulation results. Conclusion and references are presented in section 6 and 7 respectively.

2. SLAM based on EKF (Extended Kalman Filter)

2.1. The 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. Robot’s process model and observation model are nonlinear. So they need linearization by using the EKF. In other words, state transitions and measurements are rarely linear in practice. For example, a robot that moves with constant translational and rotational velocity typically moves on a circular trajectory, which can not be described by linear state transitions. This observation, along with the assumption of unimodal beliefs, renders
plain kalman filters, as discussed so far, inapplicable to all but the most trivial robotics problem.

\[ X_{k+1} = f(X_k, U_k, V_k) \]  
\[ x_{k+1} = x_k + T \left( \frac{r_w w_{k+1}}{2} \right)^2 \cos \theta_k \]  
\[ y_{k+1} = y_k + T \left( \frac{r_w w_{k+1}}{2} \right)^2 \sin \theta_k \]  
\[ \theta_{k+1} = \theta_k + \frac{r_w w_{k+1}}{I} \]  

Here, \( X_k, U_k, V_k \) are the state vector, control input, process noise respectively. \( T \) is a control time, and \( w \) and \( r \) stand for angular velocity of robot wheels.

\[ Z_{k+1} = h(X_{k+1}, M_{k+1}) \]  
\[ R = \sqrt{(x - x)^2 + (y - y)^2} \]  
\[ \phi = \tan^{-1} \left( \frac{y - y}{x - x} \right) \]  

Here, \( M_{k+1} \) is a measurement noise, \( \theta \) means robot’s heading.

Equation (2) is a motion model of general differential type robot. The function \( f \) can be used to compute the predicted state from the previous estimate and similarly the function \( h \) of equation (3) can be used to compute the predicted measurement from the predicted state. However, \( f \) and \( h \) cannot be applied to the covariance directly. Instead a matrix of partial derivatives (the Jacobian) is computed. At each timestep the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate.

The each step of extended kalman filter is as follows, Prediction step :

\[ \dot{X}_{k+1} = f(X_k, U_k, 0) \]  
\[ P_{k+1} = F_{k+1} P_k F_{k+1}^T + G_{k+1} Q_{k+1} G_{k+1}^T \]  

Here, \( Q_{k+1} \) is a covariance of process noise, \( F_{k+1} \) stands for state transition matrix through partial derivative. \( G_{k+1} \) means the control input model which is applied to the control vector.

Correction step :

\[ \dot{A}_{k+1} = Z_{k+1} - h(X_{k+1}, 0) \]  
\[ S_{k+1} = H_{k+1} P_{k+1} H_{k+1}^T + R_{k+1} \]  
\[ K_{k+1} = P_{k+1} H_{k+1}^T S_{k+1}^{-1} \]  
\[ \dot{X}_{k+1} = \dot{X}_{k+1} + K_{k+1} A_{k+1} \]  
\[ P_{k+1} = P_{k+1} - K_{k+1} S_{k+1} K_{k+1}^T \]  

where \( H_{k+1} \) is the observation model which maps the true state space into the observed space and \( R_{k+1} \) means the covariance of measurement noise.

The extended kalman filter is a recursive estimator. This means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. In contrast to batch estimation techniques, no history of observations and/or estimates is required. It is unusual in being purely a time domain filter; most filters (for example, a low-pass filter) are formulated in the frequency domain and then transformed back to the time domain for implementation.

In what follows, the notation \( X_m \) represents the estimate of \( X \) at time \( n \) given observations up to, and including time \( m \). The extended kalman filter has two distinct phases: Predict and Update. The predict phase uses the state estimate from the previous timestep to produce an estimate of the state at the current timestep. In the update phase, measurement information at the current timestep is used to refine this prediction to arrive at a new, (hopefully) more accurate state estimate, again for the current timestep.
1: Algorithm of the EKF SLAM($X_{i+1}$,$P_{i+1}$,$U_i$,$Z$)
2: $\dot{X}_{i+1} = f(X_{i+1}, U_{i+1})$
   \[ P_{i+1} = F_{i+1} P_{i} F_{i+1}^T + G_{i} Q_{i} G_{i}^T \]
3: if detected landmark
   \[ Z_{i+1} = h(X_{i+1}, M_{i+1}) \]
4: if have not seen
5: augmentation (insert landmark's position and covariance to state vector and covariance respectively)
6: else if
7: $A_{i+1} = Z_{i+1} - h(\dot{X}_{i+1}, 0)$
8: $S_{i+1} = H_{i+1} P_{i+1} H_{i+1}^T + R_{i+1}$
9: $K_{i+1} = P_{i+1} H_{i+1}^T S_{i+1}^{-1}$
10: $\dot{X}_{i+1} = \dot{X}_{i+1} + K_{i+1} A_{i+1}$
11: $P_{i+1} = P_{i+1} - K_{i+1} S_{i+1} K_{i+1}^T$
12: return $\dot{X}_{i+1}, P_{i+1}$
13: else
   back to the line 2

Table 1. Algorithm of the EKF SLAM

2.2. SLAM (Simultaneous Localization and Mapping)
The dominant approach to the SLAM problem was introduced in a seminal paper by Smith and Cheeseman in 1986, and first developed into an implemented system by Moutarlier and Chatila. This approach uses the Extended Kalman Filter (EKF) to estimate the posterior over robot pose and maps. The EKF approximates the SLAM posterior as a high-dimensional Gaussian over all features in the map and the robot pose.

Maps, in EKF SLAM, are feature-based. They are composed of point landmarks. For computational reasons, the number of point landmarks is usually small. Further, the EKF approach tends to work well in less ambiguous the landmarks are. For this reason, EKF SLAM requires significant engineering of features detectors, using sometimes artificial beacons as features.

In this simulation, we assume that there are four beacons as artificial landmarks and robot can recognize the identification of landmark. The algorithm of EKF SLAM is described table 1.

3. Robot’s Motion Model
In case of general two wheels differential type, a motion model can be described as equation (2). But a motion model of robot using caterpillar can not be made like equation (2). So we parameterize the velocity of caterpillar through real experiment in this simulation. This method makes error more large. New motion model is equation (12).

Fig. 3. Caterpillar type robot model

As see equation (12), the parameter of the control input changes to the velocity of caterpillar using encoder signal. But we have difficulty to estimate robot pose because uncertainty of motion model becomes more larger. So we propose to correct robot’s heading by using one more extended kalman filter.

4. Correct the Robot’s Heading
Robot’s heading is corrupted by its environment (surface). As stated in section 1, a robot has to estimate its pose, then the robot can build map. To correct robot’s heading, we assume that the state vector is $\theta$, and robot can correct its heading through measure absolute heading by using a gyro. So SLAM algorithm can change as follow.

1: Algorithm of the EKF SLAM($X_{i+1}, P_{i+1}, U_i, Z$)
2: $\dot{X}_{i+1} = f(X_{i+1}, U_{i+1})$
   \[ P_{i+1} = F_{i} P_{i} F_{i}^T + G_{i} Q_{i} G_{i}^T \]
3: if detected landmark
4: if have not seen
Correct heading
Table 2. Algorithm of the EKF SLAM added correction

5. Simulation Results

Simulations implemented three kinds of methods. The first is that the robot estimated own position by only odometry data. The second, the robot implemented SLAM based on the EKF, at last the simulation that is applied correction algorithm was conducted.

In Fig. 4, the green line is reference robot’s path, the black line stands for real robot’s trajectory. The red triangle and blue triangle are estimated robot pose and reference robot pose.

As you can see, when robot estimates own position by only odometry, the result is really exorbitant. Because odometry error is not only accumulated, but the robot’s heading is also corrupted by a rough surface.

Fig. 5 shows error between the reference robot’s heading and estimated robot’s heading.

Fig. 6 is the result that the robot implements SLAM based on the EKF. Compare with Fig. 4, the robot can correct own position through measurement of landmarks.

Because of landmarks, the robot corrects its pose, the heading error decrease by maximum 0.05 radian/sec to -0.03 radian/sec.

Fig. 8 is the result of SLAM based on EKF after the heading error is corrected by using one more EKF. In this simulation, the robot corrects its heading constantly by measure of the landmark.
6. Conclusion

This paper introduced supplementary EKF application to correct robot’s heading that is corrupted when the caterpillar type robot drives on a rough surface road. As comparing with three kinds of simulations, driving by only odometry data, implementation of SLAM without correcting heading and SLAM applying the proposal, this paper’s proposal is proved. When the robot moves by only odometry data, the robot can not follow reference path due to accumulate odometry error and irregular surface. In case of SLAM based on EKF, it corrects odometry error through measure landmarks, but it is not able to follow reference trajectory because of rough surface. As the robot conducts supplementary EKF for revising heading error, it can implement more accurate SLAM compare with upper two simulations. However it requires more complex computation quantity than computational complexity of SLAM based on EKF, O(N^2), because it use additional EKF to correct its heading (here, N is the number of landmarks). Futuer work is to reduce computational complexity, through experiment using real robot, proposal will have to be proved whether it is applicable or not in real environment.

8. References

A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filters for dynamic bayes nets, In Proceedings of Conference on Uncertainty in Artificial Intelligence, 2000.

Dan Simon, Kalman Filtering. Embedded Systems Programming, June, 2001.

Frank E. Schneider, Dennis Wildermuth, Using Extended Kalman Filter for Relative Localisation in a Moving Robot Formation, Fourth international workshop on robot Motion and Control, June 17-20, 2004.

Guoqiang Mao, Sam Drake, Brian D.O. Anderson, Design of a Extended Kalman Filter for UAV Localization.

G. Welch and G. Bishop. An Introduction to the Kalman Filter, ACM SIGGRAPH Tutorial, 2001.

J. Guivant and E. Nebot, Optimization of the simultaneous localization and map building algorithm for real time implementation, IEEE Transactions on Robotics and Automation, 17(3):242–257, 2001.

J. Tardos, J. Niera, P. Newman, and J. Leonard. Robust mapping and localization in indoor environments using sonar data. International Journal of Robotics Research, 2002.

M. Deans, M. Herbert, Experimental comparison of techniques for localization and mapping using a bearing-only sensor, In Proceedings of the International Symposium on Experimental Robotics (ISER), 2002.

M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Corb. A solution to the simultaneous localization and mapping (slam) problem, IEEE Transactions on Robotics and Automation, 2001.

Michael Montemerlo. FastSLAM : A Factored Solution to the Simultaneous Localization and Mapping Problem With Unknown Data Association, 2003, School of Computer Science Carnegie Mellon University.

Weyan Hu, Tom Downs, Gordon Wyeth, Michael Milford, David Fraser, A Modified Particle Filter for Simultaneous Robot Localization and Landmark Tracking in an Indoor Environment.

Shahram Rezaei, Raja Sengupta, Kalman Filter Based Integration of DGPS and Vehicle Sensors for Localization.
Stefano Panzieri, Federica Pascucci, Roberto Setola, S. Thrun, W. Burgard, D. Fox, "Probabilistic Robotics", 2005.

Simultaneous Localization and Map building Algorithm for Real-time Applications.