Comparison between Two Motion Models in EKF Localization

Sunglok Choi, JaeYeong Lee, Christiand, and Wonpil Yu
Robot Research Department, ETRI, Daejeon, Republic of Korea
{ sunglok, jylee, christiand, ywp}@etri.re.kr

Abstract—This paper examines two popular motion models in Bayesian localization: the velocity model and odometry model. Simulation and real experiment were performed to compare their accuracy in EKF localization.

Keywords—Motion Model, Velocity Motion Model, Odometry Motion Model, EKF Localization

1. Introduction

Localization is one of the most important tasks for mobile robots. Bayesian filters (e.g. Kalman filter and particle filter) are common localization technique in robotics. The Bayesian filters predict location of robot using a motion model and control input, and they correct the estimation using an observation model with sensor measurements. Thrun et al. enumerated two kinds of motion models in their book [1]: the velocity motion model and odometry motion model. The velocity model uses linear and angular velocity as control input. In contrast, the odometry model utilizes odometer measurements directly.

This paper tries to analyze and compare two motion models in EKF localization. Simulation with a virtual robot was performed in two different noise configuration. An experiment with a real robot also verified the comparison on their performance.

2. Two Motion Models

2.1 The Velocity Motion Model

In the velocity model, robot predicts its location, \( x_t, y_t \), and \( \theta_t \), using its current velocity as follows:

\[
\begin{align*}
x_t & = x_{t-1} + v_t \left[ \sin (\theta_{t-1} + w_t \Delta t) + \sin \theta_{t-1} \right] \\
y_t & = y_{t-1} - v_t \left[ \cos (\theta_{t-1} + w_t \Delta t) - \cos \theta_{t-1} \right] \\
\theta_t & = \theta_{t-1} + w_t \Delta t,
\end{align*}
\]

where \( \Delta t \) is elapsed time between \( t \) and \( t-1 \), and \( v_t \) and \( w_t \) are linear and angular velocity at time \( t \), respectively. This motion model uses radius of instantaneous curvature, \( v_t/w_t \), which entails curve displacement of robot during time interval \( \Delta t \) (Figure 1). There are several variants of the velocity model. One of them approximates curve displacement as a linear part and angular part as follows:

\[
\begin{align*}
x_t & = x_{t-1} + D \cos (\theta_{t-1} + \Delta \theta) \\
y_t & = y_{t-1} + D \sin (\theta_{t-1} + \Delta \theta) \quad (2) \\
\theta_t & = \theta_{t-1} + \Delta \theta,
\end{align*}
\]

where \( D \) is linear movement as like \( v_t \Delta t \) and \( \Delta \theta \) is angular movement as like \( w_t \Delta t \). The velocity can be measured directly by inertia sensors such as an accelerometer and gyroscope. Besides, rotary encoders on the wheels can be utilized to calculate velocity through forward kinematics. In case of a differential drive, two wheel movements derive the velocity as follows:

\[
v_t = \frac{s_R + s_L}{2 \Delta t} \quad \text{and} \quad w_t = \frac{s_R - s_L}{b \Delta t}, \quad (3)
\]

where \( b \) is distance between two wheels, and \( s_R \) and \( s_L \) are displacement of the right and left wheels during \( \Delta t \), respectively.

2.2 The Odometry Motion Model

In the odometry model, the robot directly accumulates difference of two odometer measurements as follows:

\[
\begin{align*}
x_t & = x_{t-1} + \delta_x \quad (\delta_x = \bar{x}_t - \bar{x}_{t-1}) \\
y_t & = y_{t-1} + \delta_y \quad (\delta_y = \bar{y}_t - \bar{y}_{t-1}) \quad (4) \\
\theta_t & = \theta_{t-1} + \delta_\theta \quad (\delta_\theta = \bar{\theta}_t - \bar{\theta}_{t-1}),
\end{align*}
\]

where \( \bar{x}, \bar{y}, \) and \( \bar{\theta} \) are traveled position and orientation measured by odometers (Figure 2, Red). The odometer measurement can be calculated as follows:

\[
\begin{align*}
\bar{x}_t & = \int_0^t v(\tau) \cos (\bar{\theta}(\tau)) d\tau \\
\bar{y}_t & = \int_0^t v(\tau) \sin (\bar{\theta}(\tau)) d\tau \quad (5) \\
\bar{\theta}_t & = \int_0^t w(\tau) d\tau,
\end{align*}
\]
whose discrete and recursive equation is similar with Equation 2. Two odometer measurements can be also represented as follows (Figure 2, Blue):

\[
\delta_{\text{trans}} = \sqrt{(\hat{x}_t - \hat{x}_{t-1})^2 + (\hat{y}_t - \hat{y}_{t-1})^2}
\]

\[
\delta_{\text{rot1}} = \tan^{-1}(\hat{x}_{t-1}, \hat{y}_{t-1}, \hat{y}_{t-1}) - \theta_t
\]

\[
\delta_{\text{rot2}} = (\dot{\theta}_t - \dot{\theta}_{t-1}) - \delta_{\text{rot1}}.
\]

This notation is preferred in particle filter-based localization since it is easy to model noise [1]. The odometry model with this notation becomes as follows:

\[
x_t = x_{t-1} + \delta_{\text{trans}} \cos(\dot{\theta}_{t-1} + \delta_{\text{rot1}})
\]

\[
y_t = y_{t-1} + \delta_{\text{trans}} \sin(\dot{\theta}_{t-1} + \delta_{\text{rot1}})
\]

\[
\theta_t = \theta_{t-1} + \delta_{\text{rot1}} + \delta_{\text{rot2}}.
\]

3.3 Real Experiment

An experiment was performed with a mobile robot, Starmi, and indoor GPS, StarGazer (Figure 5). Starmi adopts a differential drive platform and StarGazer detector which is a camera with IR LEDs. The detector radiates IR light toward StarGazer tags on the ceiling, and recognizes pattern of the tags. The observed tag (incorporating with its ID and relative pose) is utilized to calculate position and orientation of robot. EKF with thresholding rejected totally wrong observations which were laid out of 3\( \sigma \). Starmi moved on 6 x 6 meters rectangular trajectory from [129.4, 77.5, 0.0] T twice.

Figure 6 contains position of Starmi using EKF localization with two models. The result was similar with the first simulation due to similar magnitude of noise. Two models sustained similar accuracy at the first straight line, but the odometry model became slightly worse after the line.

4. Discussion and Conclusion

In conclusion, two motion models had similar accuracy when the sensor noise is small, but the odometry model significantly became worse when the magnitude of noise was large. This conclusion seems not intrinsic because the odometry model uses one more variable than the velocity model. One more variable means more accurate information about robot movement. Actually, the velocity model, Equation 2, is approximation of the odometry model, Equation 8, when \( D = \delta_{\text{trans}} \) and \( \Delta \theta = \delta_{\text{rot1}} + \delta_{\text{rot2}} \). However, in the experiments, EKF localization with the odometry model tended to odometer measurements, but location with the velocity model was more smooth and straight. The results mean that the additional variable gave too much information about motion (as like overfitting in machine learning).
Fig. 3. Simulation Result #1 (Location Sensor Noise: $\sigma_x = \sigma_y = 0.1$ and $\sigma_\theta = 3$ degrees)

Fig. 4. Simulation Result #2 (Location Sensor Noise: $\sigma_x = \sigma_y = 0.3$ and $\sigma_\theta = 10$ degrees)

Fig. 6. Experimental Result (A Robot Starmi and Location Sensor StarGazer)

Acknowledgement

The authors would like to thank Haksoo Lee (Hanool Robotics Co.) for supporting StarGazer.

References